电脑

7
1987

中国软件行业协会会刊





廣東省計算中心的試驗車間，可生產各種電腦接口板、中英文打字機、新型收音機等電子產品。

GCC' s workshop can produce the computerised Chinese-English typewritters , new model radio sets , computer boards and another kinds of electronic products .

技術貿易部備有多種電腦模板、易耗品、常用備件，各種軟件，爲計算機用戶提供方便。

Technical Trade Group is ready to supply the parts , boards , consumer goods and software products to the customers .





廣東省計算中心重視與國內外同行的聯系和合作，擧行各種研討會，交流會，學習和借鑒國外技術，和國外廠商進行合作經營。

GCC constantly focuses its attention on the close ties and co-operation with the companies of same trade or occupation abroad and at home . A lot of seminars and meetings have been held for technology exchange . GCC is looking for an opportunity to co-operativly run business with the foreign firms and manufacturers .

中心多次派人到國外、港澳地區進行考察、培訓、進修和攻讀博士、碩士學位。

For recent years GCC has sent numbers of persons to go abroad and to H . K and Macao to make an investigation , to take training and to engage in advanced studies to earn Ph. D and Master degrees .



廣東省計算中心擁有 IBM 4341—2 大型機系統、數十台各種類型的微型機，并配有豐富的軟件。向社會開放，爲各界提供服務。右圖爲大型機機房一角。

In GCC there are a mainframe IBM 4341-II , tens of various microcomputers and numbers of softwares which provide computer service for public use . Right is the section of the machine room.



廣東省計算中心已有一大批計算機系統硬件、系統軟件開發、維護人員，應用軟件開發人員。幾年來在各個方面取得了成果與經驗。

Numbers of technical persons are working for GCC to develop the computer systems and the application softwares and to maintain them . For recent years they have made achievements and gained experience in many R&D areas .



跟踪國際先進技術，本中心工程技術人員積極開發計算機网絡、計算機輔助設計（CAD）等方面的應用。

For keeping up with the latest technology in the world , the technical personnel are actively developing the applications in the computer network and CAD fields .



廣東省計算中心根据我國國情，按照軟件工程開發規范。掌握高水平的開發技術，使軟件開發的質量得到保証。右圖爲省級機關辦公室自動化課題組在調試程序。

The software engineering philosophy and technics are now studied and used in GCC for developing the high-qaulity software products . The development team of the OA system for a provincial organization is debugging the programs right .





幾年來，廣東省計算中心先後開發了多種漢字計算機系統。爲我國的計算機應用作出了貢獻。左圖爲中心研製開發的縮微機控製電腦漢字系統。

For recent years GCC has developed many kinds of the microcomputers with Chinese function and thus made a contribution to computer applications in China.

Chinese microcomputer left for controlling the microfilm machine is the one of such results.



電腦用於生產過程控製，投資少、經濟效益高。廣東省計算中心積極開拓單片機在過程控製、智能儀器、儀表中的應用。左圖所示的單片機煮糖儀體積小、價格低，可靠性高，操作方便。

The industry control projects to be realized by micros have high rate of benefit/cost. GCC' s developers have focused much of their effort on this area. The left picture shows that the single-chip proccessor is used in sugar-boiled procss. it is of small size, lower price, high reliability and easy-to-use.



維修服務部爲國外、港澳地區多家公司設有維修站。左圖爲維修人員正在熟練地檢測計算機。

Some computer service centers committioned by the firms from foreign countries, H. K and Macao are attached to the GCC' s Technical Service Department. The picture shows that the service staff members are skillfully examining and repairing the broken microcomputer.



廣東省計算中心開辦計算機大專班，各種計算機培訓班，爲各地區、各部門培養計算機專業人才和技術更新。

The special college classes attached to GCC have been opened and various training classes have been held by GCC to bring up the qualified persons in the computer field and to renew technology for the sectors in the industry.

电脑

（月　刊）

1987年

第七期（总第19期）

电脑应用

IBM－PC／XT机与SR－6602绘图仪的联机应用体会…………王全峰　田立坚（3）

大容量dBASEⅢ数据拷入硬盘…………邢　进（5）

文件名加密保护的两种解密法…………穆卫东（6）

微机运用于企业干部的考核…………宋士倩（8）

用BASIC程序来管理WORDSTAR文件——介绍一个“中文阅读程序”…………朱树明（11）

软件纵横

机器语言子程序的连结方式…………熊锡义（12）

利用微机的自动计时功能建立微机工作数据库…………许传国（14）

APPLE SOFT语言程序复盖技术初探…………陈　昕（16）

如何在DOS3.3上使用35条以上磁轨…………刘志仁（19）

dBASEⅢ PLUS数据管理标准命令语法…………彭荣昌　陈锦荣（21）

大学生之页

使用IBM－FORTRAN语言的几点体会…………柳　影　王　健（24）

设计安装利用微机测量直流电压的简易接口机…………伍志峰（26）

中学天地

验证角谷猜想…………杨克昌（4）

二元一次方程组练习题程序…………周慕贤（13）

怎样提高APPLEⅡ机的计算精度…………赵政文（17）

在LASER310上使用ON语句…………施国范（20）

一行程序设计技巧一谈…………邱小宁（25）

条件式的功能与应用…………陈庆祥（29）

APPLEⅡ绘图技巧…………谢志坚（31）

COMX PC1 BASIC中CHR $函数的妙用…………马　剑（33）

打印上下标的一种方法…………………………………………… 蔡希人（35）
程序优化一例…………………………………………………… 杨　纯（40）

电脑用户

关于dBASEⅢ子串函数的巧妙用法…………………………… 邓晓琼（36）
电脑的选购、验机和投入使用要点…………………………… 陈庆章（38）
运用程序语言处理汉字信息的一些方法……………………… 陈汝祥（41）
用APPLEⅡ微机检测噪声中微弱信号…………………… 秦卓仁　崔如春（44）

万花筒

微电脑演奏乐曲………………………………………………… 王新民（48）
电脑艺术（1）…………………………………………………… 秦华山（28）
电脑艺术（2）…………………………………………………… 李　丹（34）

厂家与产品

广东省计算中心简介…………………………（封面、封底、封二、封三）

简讯

顺德县电脑应用形势喜人……………………………………………（7）
中国软件行业协会DG应用协会首届年会在苏州举行………… 钱承德（37）
缩微检索投影仪及其联机系统研制成功……………………… 宋　川（43）

责任编辑：曾令宝　吴军　柯庆
美术编辑：丘文东

主　　办　电子工业部
编　　辑　《电脑》编辑部
（地址：广州市石牌华南师范大学微电子学研究所）
出　　版　华南师范大学微电子学研究所
排　　版　电脑杂志社电脑激光排版中心
印　　刷　粤北印刷厂
总发行处　韶关市邮电局
订 阅 处　全国各地邮电局、所
刊　　号　46—115
定　　价　0.50元
出版日期　87年7月1日

广东省期刊登记证第118号
广告经营许可证穗工商广字004号

IBM—PC／XT机与SR—6602绘图仪的联机应用体会

沈阳工业大学　王全峰　田立坚

在此，笔者想就在IBM—PC／XT机与SR—6602绘图仪的联机应用过程中所遇到的问题及处理的办法浅谈一下体会，旨在相互交流和学习。

一．问题

1986年我们从日本订购一台SR—6602绘图仪，用来与IBM—PC／XT机联机进行工程设计计算和绘图。但在联机过程中，在Basic 2.0版本系统状态下，无论是用并行口还是串行接口，SR—6602绘图仪都不动作，而在同一系统下与日本产671—20绘图仪联机，用并行口或串行接口都获得成功，又在AUTOCAD 2.1状态下，与SR—6602绘图仪用串行接口联机时也获得成功。一时间我们不知问题的所在。

二．寻找原因及处理

经过分析我们发现，AUTOCAD 2.1能够支持SR—6602绘图仪，这表明我们订购的绘图仪本身没有问题；而Basic 2.0版本系统能够支持日本产671—20绘图仪，这说明问题可能发生在Basic系统与SR—6602绘图仪之间，即SR—6602绘图仪要哪一级以上的Basic版本来支持它，和SR—6602绘图仪如何接收PC机的支持，于是我们将Basic系统换为2.01以上的版本再进行联机，此时无论是用并行接口还是串行接口，联机都获得成功，由此说明SR—6602绘图仪是需要Basic 2.01以上的版本的支持。

三．联机

若绘图仪与IBM—PC／XT机用并行接口联机，则可将SR—6602绘图仪通过电缆插到打印机接口上，用LPRINT语句发出绘图命令，即LPRINT“绘图命令字符串”，此时打印机与绘图仪当然不能同时使用。如欲同时使用，应增加Centronix并行接口扩展板。如果使用RS—232串行接口绘图仪，则IBM—PC／XT微型机也应使用RS—232接口与绘图仪相连，计算机和绘图仪的传送格式和速率的微型开关必须设置的相互一致。

下面的程序是我们在联机过程中用来画曲线的调试小程序及结果，程序一用于并行接口，程序二用于RS—232串行接口。

以上所述对IBM—PC／XT的兼容机同样适用。

程序一

```
5 'FOR PARALLEL CONNECTOR DRAWING PROGRAM '
10 OPEN "COM1 :  8N82" AS #1
15 GOTO 35
20 PRINT P$
25 LPRINT P$
30 RETURN
35 P$="EX;a":GOSUB 20
40 P$="EB;&":GOSUB 20
45 P$="IN&AP; 0,1000,-3": GOSUB 20
50 P$="AP;1050,0, 3&AP;100,0,2":GOSUB 20
55 FOR,X=-22 TO 22'STEP .19
60 Y=400*SIN(X)/X
65 S1=INT(X*20) +560: S2=INT(Y*1.5)
70 P$="AP;" +STR$(S1)+","+STR$(S2)+",2"
75 GOSUB 20
80 NEXT X
85 P$="AS; 50,400, 8,0, Y=SIN(X)/X"
90 GOSUB 20
95 P$="TF":GOSUB 20
100 END
```

程序二

```
10 'FOR SERIES CONNECTOR DRAWING PROGRAM '
15 OPEN "COM1:9600,e,7,1,cs,ds65535,cd" AS #1
20 GOTO 40
```

验证角谷猜想

湖南娄底师专计算机室　杨克昌

在大量实践基础上人们猜想：从任意正奇数出发，反复作以下两种运算

(1)、若为奇数，则3倍加1

(2)、若为偶数，则除以2

最后都会得到数1。

例：3→10→5→16→8→4→2→1

这一猜想先流行于美国，十几年前由角谷静夫教授把它介绍到日本，通常把这一猜想称为“角谷猜想”。至今，还没有人能证明或推翻这一猜想。我们试编一个BASIC程序让电子计算机对这一有趣的猜想作有限次验证。

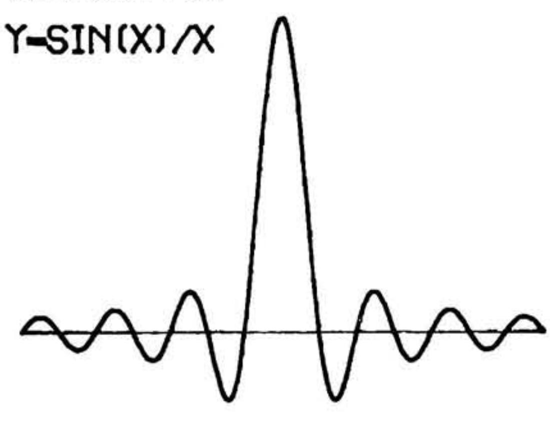
程序及在APPLE II上键入T＝100（即验证100以内的正奇数）的运行结果附后：

打印结果中每一奇数后括号内数字为得到1的运算次数。如3（7）表示数3要经7次运算才能得到数1。

如果需显示所要验证的奇数运算过程的中间结果，只要在上述程序中加几个打印语句即可。

```
5   INPUT "T=";T
10   PRINT "3---";T
15   FOR N = 3 TO T STEP 2
20  M = N:S = 0
30  M = M * 3 + 1:S = S + 1
40  M = M / 2:S = S + 1
50   IF M = 1 THEN 80
60   IF M / 2 =  INT (M / 2)
     THEN 40
70   GOTO 30
80   PRINT N;"(";S;")",
90   NEXT N
100  END
]RUN
T=100
3---100
3(7)        5(5)        7(16)
9(19)       11(14)      13(9)
15(17)      17(12)      19(20)
21(7)       23(15)      25(23)
27(111)     29(18)      31(106)
33(26)      35(13)      37(21)
39(34)      41(109)     43(29)
45(16)      47(104)     49(24)
51(24)      53(11)      55(112)
57(32)      59(32)      61(19)
63(107)     65(27)      67(27)
69(14)      71(102)     73(115)
75(14)      77(22)      79(35)
81(22)      83(110)     85(9)
87(30)      89(30)      91(92)
93(17)      95(105)     97(118)
99(25)
```

```
25 PRINT #1,P$
30 PRINT P$
35 RETURN
40 P$="EX;A":GOSUB 25
45 P$="EB;&":GOSUB 25
50 P$="IN&AP; 0,1000,-3": GOSUB 25
55 P$="AP; 1050,0, 3&AP;100,0,2":GOSUB 25
60 FOR X=-22 TO 22 STEP .19
65 Y=400*SIN(X)/X
70 S1=INT(X*20)+560: S2=INT(Y*1.5)
75 P$="AP;" +STR$(S1)+","+STR$(S2)+",2"
80 GOSUB 25
85 NEXT X
90 P$="AS; 50,400, 8,0, Y=SIN(X)/X"
95 GOSUB 25
100 P$="TE":GOSUB 25
105 CLOSE #1 : END
```



大容量DBASE Ⅲ数据拷入硬盘

四川绵阳二O七信箱　邢　进

在IBM PC／XT或IBM PC／AT及其兼容机上建立了大容量的dBASE数据文件后，可以从硬盘（在机内的驱动器）把数据拷贝出备份文件保存在软盘上以便送档或携带。例如，我们研制的设备管理软件系统，其中机动设备数据库文件的记录有三千多个，把这些数据从硬盘中拷贝出，需要两张5.25英寸软盘才能存放下。

《软件报》第8期（86.4.16）第二版登载了一篇题为《dBASE Ⅲ的硬盘数据备份》的文章介绍了如何把大容量数据备份问题。相反，在实际工作中会遇到把大容量（大于360KB以上）数据库拷人硬盘，这就出现了：如何将大容量（两软盘以上，每软盘容量：360KB）dBASE Ⅲ数据拷入同一数据库内的问题。

为了解决这个问题，本人编制了一个实用程序（程序附后）。

运行本程序时，操作者需输入要建立到硬盘上的数据库名称。软盘数量及插入数据软盘的驱动器名，就可以进行拷贝。

说明：

（1）本程序与《软件报》第8期（86.4.16）第二版程序配套使用。

（2）定义建在硬盘的数据库名必须与备份软盘上库名一致。

（3）插入软盘的驱动器只能是A或B。

（4）拷贝顺序是：先拷得备份软盘数据，后拷入硬盘。

（5）软盘数据不能重复拷入。

本程序具有汉字提示，使用时按提示操作。使用方便，可在IBM—PC及兼容机上使用。本程序在IBM—PC／XT机上通过实际使用。

```
2p  1p
** KB.prg
set talk off
stor space(1) to drive
stor space(8) to filename,filen,kmc
stor 0 to diskcount
do while .t.
 clear
 @ 1.10 say 将 软 盘 数 据 库 拷 入 硬 盘
 @ 2.10 say '======================='
 @ 4.10 say '拷 入 硬 盘 数 据 库 名 定 义
 为 :   get filen
 read
 stor trim(filen) to filename
 if file("C:&filename..dbf")
 clear
stor "  " to ss
 @ 4.10 say '是 否 覆 盖 原 数 据 库 ?
 是 ( Y ) , 否 ( N ) ' get ss
read
if upper(trim(ss))="Y"
exit
endif
*stor space(8) to filen
*p
 else
 exit
 endif
enddo
do while .t.
 clear
 @ 4.10 say '输 入 拷 贝 使 用 驱 动 器
 名 : ' get drive
 read
 stor upper(drive) to drive
 if drive="A" .or. drive="B"
 exit
 else
 clear
 @ 4.10 say '输 入 错 误 , 只 能 是 A 或
  B 驱 动 器'
wait '按 任 一 键 . 重 新 输 入 !'
 loop
 endif
enddo
stor 1 to m
do while .t.
*clear
*p
@ 1.6 say '请 把 第 : '+str(m,2)+' 号 软
盘 插 入 '+drive+' 驱 动 器 内'
@ 3.10 say '软 盘 数 据 库 名 为 (
 '+filename+'.'+str(m.2)+' ) '
@ 5,10 say '  '
 wait '        按 任 一 键 开 始 拷 贝 !'
 clear
 @ 4.10 say '正 在 拷 贝 第   '+str(m.2)
 +' 号 软 盘  ......'
stor drive+":"+filename+"."+str(m.1) to tha
if m=1
 use &tha
 copy to c:&filename
 else
use c:&filename
set talk on
append from &tha
set talk off
endif
stor m+1 to m
clear
stor '  ' to df
*@ 4.6 say '是 否 有 第   '+str(m.2)+' 号 软
*p
 盘 ?    有 ( Y ) , 否 ( N )   get df
read
if upper(trim(df))<>"Y"
exit
endif
enddo
clear
@ 4.20 say '全 部 数 据 软 盘 拷 贝 完 !'
*return
*p
```

文件名加密保护的两种解密法

成后驻昆明办事处指挥自动化室　穆卫东

《计算机世界》去年第六期，刊登了题为《简便的汉字码文件名保密方法》一文，介绍了在文件名中插入汉字码，以造成文件名的表里不一，从而达到对文件的保密。此法由于简单方便，现已被众多的软件人员所采用。基于这种情况，所以，目前对于如何破译这种保密法已使越来越多的计算机人员发生兴趣。在此，本文介绍两种原理简单，操作方便的破密法，以供读者参考!

一、用DEBUG解密

这种方法我们又将它称之为“真破”。何为“真破”，即我们并不去修改它的插在文件名中的半汉字码，而是只要知道了该码的实际值，然后采取某种手段输入即可。很明显，在计算机内的一个字节，不管其高位是否为1，均有一个与之对应的十六进制值。这一点，用DEBUG可以看到。在这里，我们不妨重提一下，在PC系列机内，其ASCII码是从0～255的，即大于127（高位为1），同样有其实际意义。例如：“1/2　”的ASCII码值是171；“1/4”的ASCII码值是172等等。对于这种大于127（高位为1）的ASCII码字符，系统同时提供了其键盘输入方法。在CCDOS中，两个高位为1的内码方能形成一个汉字。显然，对于单个的汉字码，我们仍然可按其为大于127（高位为1）的ASCII码对待。并利用DOS提供的输入方法输入之。问题即可迎刃而解!

下面以实例说明：

例：A盘上有文件用单个汉码加密。运行DEBUG，将A盘文件目录区用L命令调入从0100开始的内存区域中，用D命令显示如下：

```
DEBUG
-L 0100 0 5 1
-D
55E8:0100  45 52 52 4F 52 20 20 20-50 52 47 20 00 00 00 00   ERROR   PRG ....
55E8:0110  00 00 00 00 00 00 30 03-21 00 02 00 80 00 00 00   ......F.!.......
55E8:0120  54 54 C8 54 54 20 20 20-42 41 53 20 00 00 00 00   TTHTT   BAS ....
55E8:0130  00 00 00 00 00 00 B5 09-21 00 03 00 80 0D 00 00   ......5.!.......
55E8:0140  43 43 51 2D 32 53 59 31-50 52 47 20 00 00 00 00   CCQ-2SY1PRG ....
55E8:0150  00 00 00 00 00 00 36 0B-21 00 07 00 00 02 00 00   ......6.!.......
55E9:0160  4E 41 4D 45 B0 20 20 20-50 52 47 20 00 00 00 00   NAME0   PRG ....
55E8:0170  00 00 00 00 00 00 4E 0E-21 00 08 00 80 03 00 00   ......N.!.......
-Q
```

可以看到，文件TT　TT.BAS，NAME　.PRG是被加密的。其中插入的半汉字码的值分别是：TT　TT.BAS文件为C8H；NAME　.PRG文件为B0H。需要提醒的是：右边的ASCII码字符显示，两密码字符分别为H和0，但在此切不可将其以H和0的ASCII码计算，中间显示的十六进制值才是该码的实际值。这一点应特别加以注意。知道了两码的十六进制值之后，分别将其化为相应的十进制值。结果是：C8H=200；B0H=176；然后用Q命令退出DEBUG。进行到此，可以说问题已经基本解决，但在实际应用中，这两码又是如何输入？DOS提供的大于127的ASCII码的输入方法是：按下中间键盘左下方的Alt键，然后再敲右边小键盘该码的十进制值，则该码即被输入。现在如我们需要在DOS状态下显示文件TT　TT.BAS的内容，具体操作如下：

A>TYPE　TT（TT键入后，按下Alt键，然后再敲右边小键盘200）TT.BAS［回车］。（紧接着再键入TT.BAS，回车）。则该文件内容即被显示在屏幕上。

```
A>TYPE  TT  TT.BAS
10  CLS
20  PRINT  " *****密码破译*****"
30  END
```

文件NAME　.PRG亦同，在此不再重复。

由于这种解密法并未改动原文件名，而是通过一系列方法键入该密码。所以称之为“真破”。

二、用COPY命令解密

DOS中的内部命令COPY是用来复制文件的。它的功能较强。仔细分析一下，可以看到，它有两个特点：

1. 命令中可含有“？”（代替一个任意ASCII码字符）和“*”（代替多个任意ASCII码字符）。

2. 在文件复制的同时，可进行文件名更改。

掌握了COPY命令的这两个特点之后，我们即可将其充分运用到解密中来。仍以上面两个文件为例，操作如下：

```
A>
A>COPY  TT?TT. BAS  ABCDE. BAS
TT  TT.BAS
```

1个文件已被拷贝

在复制的同时，以“？”代替密码符，同时将文件名改为ABCDE. BAS。

```
A>
A>COPY  NAME*. PRG  MWD PRG
NAME  .PRG
```

1个文件已被拷贝

复制的同时，以“*”代替密码符，且将文件名改为MWD. PRG。

不言而喻，其后对该文件的所有操作，使用新名字即可。

当然，“？”和“*”号的应用，完全可以灵活掌握。由于这种解密法并不需要知道其密码是多少，而是采用“？”和“*”号避开的方法，同时亦更改了原文件名，故称之为“假破”。

用COPY命令解密，需要注意以下两点：

1. 新文件名应尽量避免少用或不用与老文件名中相同的字符。

2. 在一块盘上或在一个目录下，使用“？”和“*”号所涉及的文件最好单一。

以上是我们在进行文件名保密法解密研究中归纳出的两种解密法，不妥之处，敬请指出！

顺德县电脑应用形势喜人

顺德县现有各种微型计算机系统200多套，有VAX—11/730小型机1台，IBM—PC及长城0520系列约占20%，大量为APPLE—II微机；除分布于学校，金融管理机构、科研、生产企业等部门外，不少个人亦购机自用。上述情况还未包括工业设备中用于检测与控制系统（如数控机床、数控切割机自动编程等系统的近百台微机）。全县从事电脑应用专业工程技术人员四十多人和二百多名操作员。近年来县里先后举办电脑应用技术在职人员培训和BASIC课程十四个班，参加基础学习有五百多人次。

县政府一九八六年专门拨出几十万圆巨款建立起“顺德县电脑培训服务中心”，加速人才培训，致力电脑应用技术的推广和普及，积极开展技术交流，软件开发与加强社会技术服务。接着一九八七年一月成立了“顺德县计算机应用技术学会”，第一批会员绝大多数是现职计算机技术骨干，县委、县政府、县科委、科协和不少政府部门领导人以及省、市计算机学会负责人均出席了成立大会，并表示一定大力支持和促进学会的工作。

为了加强科技与生产的结合，一九八七年二月，成立了“西顺电子技术开发中心”。这是由县科委、顺德县经济贸易总公司、西北电讯工程学院三方共同联合组成的经济技术实体。该机构成立后已陆续承担了计算机技术应用的硬件、软件开发，工业自动控制，电子新产品开发，技术咨询，方案论证，人才培训，引进技术的消化、吸收与工程安装等等一批技术工程项目。　(陈仲良)

微机运用于企业干部的考核

华南计算机公司计划处　宋士倩

在现代企业中，管理干部的水平和组织效能，往往是企业最重要的资源之一。过去人们都是运用定性的方法来处理人员的考核，往往难以跳出人为因素过重的框框。现代科学的发展，打开了人们的思路，许多具有科学性，把定性考核建立在定量考核基础上，尽量反映出干部真实面貌的方法脱颖而出。本文在此提出一种从系统论的观点出发以模糊数学的综合评判为根据，以微机为工具对企业干部进行考核的科学方法。

一.方法概述：

考核一个企业的管理干部应从思想素质、管理能力、学识水平、工作成就这四个方面来考虑，而其中每一个方面又分别受到一系列因素的影响。因此，首先成立一应答小组，小组的成员分别是被考核者的上级、平级、下级，小组成员将对被考核者就以上提出的因素打分，根据打分者的地位取一定的权数，从而得出一个评价矩阵，又根据这些因素在各个方面所起的作用程度便得出一个权数矩阵。程序便利用综合评判的方法，对评价矩阵与权数矩阵进行模糊变换，则可得出被考核者在以上四个方面应得的分数，这四个方面对被考核者的总分的影响程度又组成了一个权数矩阵，四个方面的得分便组成了一个新的评价矩阵，通过又一次的模糊变换便得出被考核者的总得分。我们将这五个数据描绘在一个坐标上，那么一个人的素质就被直观地描绘出来了（如图1）。这个结果能综合一个人的素质，比个人感情用事的评价要准确得多。另外从图中可知：M_1、M_2、M_3、M_4 素质四个方面的得分。点M_i（$i=1\sim4$）的位置越高说明该方面越好，M_5 的高低就反应一个人的总得分的情况，若将M_1，M_2，M_3，M_4连接起来，该曲线的高低位置就反应出人的素质的好坏。因此在用人，提拔干部时就可以从中选择合适的人选，使其充分发挥自己的特长，使工作做得更好。

二.使用方法

1. 由被考核者的上级、同级、下级各若干人组成一应答小组，就下列的表格（表1）进行打分，每一方面由五小栏组成，每栏实行5分制。

2. 根据与被考核者的关系将其分成三类。按上级，同级，下级顺序，将其打分结果输入机中，在输入过程中，每输完一张表，就必须回答下张表是否与此表同类。若是则回答“0”，不是则回答“1”。

3. 当三类人的表格输完以后，打印机将打出被考核者的评价矩阵，各个方面的得分及总分，并在坐标上反映出来。

三.举例：

现有10个人（其中上级2人，同级2人，下级6人）对某人打分，其结果见程序570～765句，其权数矩阵，一级评判的矩阵见程序75～90句，二级评判的矩阵见程序70句，评判结果见程序后面的打印结果：$S=3.46$，$M_1=4.33$，$M_2=3.806$，$M_3=4.125$，$M_4=3.667$。

四.说明：

1. 权数矩阵可以由考核单位根据本单位的工作性质，方式而一一确定。

2. 程序中的分数读人可以用“INPUT”语句，为了显示方便，本文用“READ，DATA”语句。

3. 本程序在APPLE Ⅱ上通用，并在攀钢人事处试用过。

4. 由于篇幅有限，其程序中的算法，数学模型就不再一一叙述。

```
]LIST

10  REM  THIS PROGRAM IS FOR CAR-
     DS MANAGEMENTING
15  DIM C(4),E1(5),R2(4)
20  DIM A1(4,100),A2(5,5)
25  DIM R1(4,5),A3(4,5)
27  DEF  FN H(Q) =  INT (Q * 1000
     ) / 1000
```

```
30   FOR J = 1 TO 4
35   READ R2(J)
40   NEXT J
45   FOR J = 1 TO 4
50   FOR I = 1 TO 5
55   READ R1(J,I)
60   NEXT I
65   NEXT J
70   DATA  0.2,0.2,0.3,0.3
75   DATA  0.25,0.2,0.2,0.2,0.15
80   DATA  0.25,0.25,0.2,0.15,0.15

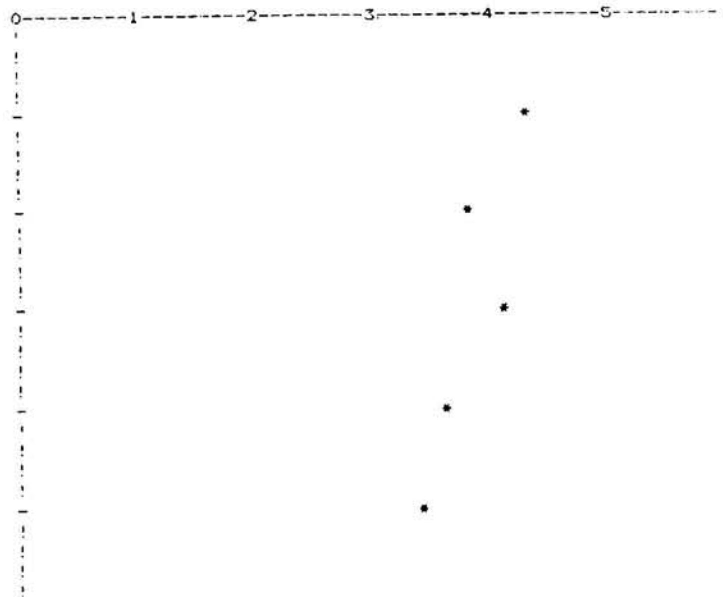
85   DATA  0.3,0.2,0.2,0.2,0.1
90   DATA  0.3,0.3,0.2,0.1,0.1
95   FOR T1 = 1 TO 3
100 C(T1) = 0
102  FOR J = 1 TO 100
105 A1(T1,J) = 0
110  NEXT J
130  FOR J = 1 TO 4
135  FOR N = 1 TO 5
140  READ X
141  IF X > 5 THEN 144
142  IF X < 1 THEN 144
143  GOTO 146
144  PRINT "INPUT WRONG !! REINPU
     T:"
145  GOTO 140
146 T2 = X + (N - 1) * 5 + (J - 1
     ) * 25
150 A1(T1,T2) = A1(T1,T2) + 1
155  NEXT N
160  NEXT J
165 C(T1) = C(T1) + 1
170  PRINT "ANY DATA ABOUT THE SA
     ME KIND OF PERSON--Y(0)/N:"
175  INPUT S
180  IF S = 0 THEN 130
185  NEXT T1
190  REM
195 C(4) = 1.5 * C(1) + C(2) + 0.
     5 * C(3)
200  FOR J = 1 TO 100
205 A1(4,J) = (1.5 * A1(1,J) + A1
     (2,J) + 0.5 * A1(3,J)) / C(4
     )
210  NEXT J
220  FOR J = 1 TO 4
225  FOR I = 1 TO 5
230  FOR N = 1 TO 5
235 T2 = (J - 1) * 25 + (I - 1) *
     5 + N
240 A2(I,N) = A1(4,T2)
245  NEXT N
246  NEXT I
247  PRINT
248  PRINT
251  FOR D1 = 1 TO 5
252  FOR D2 = 1 TO 5
253  HTAB (D2 * 6): PRINT  FN H(A
     2(D1,D2));
254  NEXT D2
255  PRINT
256  NEXT D1
259  FOR I = 1 TO 5
260 M = 0
265  FOR N = 1 TO 5
266 F = R1(J,N)
267  IF R1(J,N) < A2(N,I) THEN 26
     9
268 F = A2(N,I)
269  IF M > F THEN 275
270 M = F
275  NEXT N
280 E1(I) = M
285  NEXT I
290 M = 0
295  FOR I = 1 TO 5
300 M = M + E1(I)
305  NEXT I
310 C(J) = 0
315  FOR I = 1 TO 5
320 A3(J,I) = E1(I) / M
325 C(J) = C(J) + I * E1(I) / M
330  NEXT I
335  NEXT J
340  FOR J = 1 TO 5
345 E1(J) = 0
350  FOR I = 1 TO 4
351 F = R2(I)
352  IF A3(I,J) > F THEN 354
353 F = A3(I,J)
354  IF E1(J) > F THEN 360
355 E1(J) = F
360  NEXT I
365  NEXT J
366  FOR D1 = 1 TO 4
367  PRINT
368  FOR D2 = 1 TO 5
369  HTAB (D2 * 6): PRINT  FN H(A
     3(D1,D2));
370  NEXT D2
371  NEXT D1
372 M = 0
373  PRINT
375  FOR J = 1 TO 5
380 M = M + E1(J)
385  NEXT J
390 S = 0
395  FOR J = 1 TO 5
400 S = J * E1(J) / M + S
405 E1(J) = E1(J) / M
410  NEXT J
415  PRINT "***************LAST RS
     ULTS:*****************"
416  PRINT "      S="; FN H(S)
420  PRINT "     M1=";C(1)
425  PRINT "     M2=";C(2)
430  PRINT "     M3=";C(3)
435  PRINT "     M4=";C(4)
436  PRINT
437  PRINT "*************DIAGRAM:*
     *************"
438  PRINT : PRINT : PRINT
440 D4$ = "---------"
450  FOR D1 = 0 TO 5
455  PRINT D1;D4$;
460  NEXT D1
465  PRINT ">"
470  FOR D1 = 1 TO 4
475  FOR D2 = 1 TO 4
480  PRINT "!"
485  NEXT D2
486 W =  INT (C(D1) * 10)
487  PRINT "-";
490  FOR I = 1 TO W - 1: PRINT "
     ";: NEXT I
494  PRINT "*"
495  NEXT D1
546  FOR I = 1 TO 4
548  PRINT "!"
550  NEXT I
552  PRINT "-";
553 W =  INT (S * 10)
554  FOR I = 1 TO W - 1
555  PRINT " ";: NEXT
556  PRINT "*"
557  PRINT "!": PRINT "!": PRINT
     "!": PRINT "!"
560  PRINT "     *********SUCCESS**
     *******"
564  PR# 0
565  END
570  DATA  5,4,5,5,4
```

```
575  DATA  4,5,4,4,4
580  DATA  5,4,3,5,4
585  DATA  4,5,5,3,1
590  DATA  4,4,5,5,4
595  DATA  4,5,4,4,3
600  DATA  4,4,3,5,4
605  DATA  4,5,4,3,1
610  DATA  5,5,5,5,5
615  DATA  5,5,4,4,4
620  DATA  5,4,3,5,4
625  DATA  4,5,5,3,1
630  DATA  4,4,4,5,4
635  DATA  4,4,3,3,3
640  DATA  4,4,3,5,3
645  DATA  4,5,4,3,1
650  DATA  5,5,5,5,5
655  DATA  5,5,4,4,4
660  DATA  4,4,4,5,4
665  DATA  5,5,4,3,2
670  DATA  5,4,5,5,4
675  DATA  5,5,4,4,3
680  DATA  4,4,4,5,4
685  DATA  5,5,4,3,2
690  DATA  5,4,5,5,4
695  DATA  5,5,4,4,3
700  DATA  4,4,3,5,4
705  DATA  5,5,4,3,1
710  DATA  4,4,4,5,4
715  DATA  5,5,4,3,3
720  DATA  4,4,3,5,4
725  DATA  5,5,4,3,1
730  DATA  4,4,4,4,4
735  DATA  4,5,4,3,2
740  DATA  4,4,3,4,4
745  DATA  4,5,4,3,1
750  DATA  3,4,4,4,3
755  DATA  3,5,4,3,2
760  DATA  4,4,3,4,4
765  DATA  4,5,4,3,1

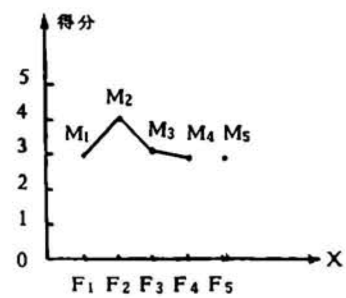
]RUN
ANY DATA ABOUT THE SAME KIND OF PERSON--Y(0)/N:
?0
ANY DATA ABOUT THE SAME KIND OF PERSON--Y(0)/N:
?1
ANY DATA ABOUT THE SAME KIND OF PERSON--Y(0)/N:
?0
ANY DATA ABOUT THE SAME KIND OF PERSON--Y(0)/N:
?1
ANY DATA ABOUT THE SAME KIND OF PERSON--Y(0)/N:
?0
ANY DATA ABOUT THE SAME KIND OF PERSON--Y(0)/N:
?0
ANY DATA ABOUT THE SAME KIND OF PERSON--Y(0)/N:
?0
ANY DATA ABOUT THE SAME KIND OF PERSON--Y(0)/N:
?0
ANY DATA ABOUT THE SAME KIND OF PERSON--Y(0)/N:
?0
ANY DATA ABOUT THE SAME KIND OF PERSON--Y(0)/N:
?1

***************LAST RSULTS:******************
     S=3.46
    M1=4.33333334
    M2=3.80645161
    M3=4.125
    M4=3.66666667

*************DIAGRAM:**************
```




```
********SUCCESS*********
```



图一

表一，

被考核者姓名															与被考核者关系，					
考核方面	思想素质					学识水平					管理能力					工作成绩				
考核要素	政治态度	工作作风	道德品质	责任感	群众关系	管理知识	专业知识	技术或业务水平	知识范围	基本经济理论	决策与果断	用人与组织水平	创造能力	工作效率	其它能力	工作数量	工作质量	突出成绩	各级表彰	其它贡献
得分																				

用BASIC程序来管理WORDSTAR文件

——介绍一个“中文阅读程序”

湖北省孝感市39155部队自动化工作站　朱树明

现在许多单位都用微机来贮存、检索资料。常用的方法大致分为两种，一是直接将汉字存于BASIC程序的PRINT语句或DATA语句中，这种方法便于实现程序控制，使用起来比较灵活，但因对文件的编辑功能较弱，编排比较繁琐，不适于大量资料的贮存。另一种办法就是用最常用的中文编辑软件WORDSTAR进行资料的存贮，利用WORDSTAR强有力的汉字编辑功能能很方便地进行资料的编排、存贮和修改，这是用得较多的一种方法，但用这种方法贮存的文件不便于调看、阅读。计对这种情况，笔者专门编制了一个“中文阅读程序”，用一个BASIC程序将WORDSTAR编辑的文件管理起来，对于阅读文本文件非常方便、直观。

为便于阅读，本程序提供两种显示方式，即大汉字方式（每行40汉字，中分状态下）和小汉字方式（每行80汉字，高分状态下）。每屏显示10行，翻页控制可为手控或自动定时循环。

处理方法：若文本中一行不超过80（或40）个汉字则直接“读出”显示，若一行超过80（或40）个汉字，则截留前80（或40）个汉字，余下汉字分配到下一行，与一下行原来前面的汉字“合并”，凑够80（或40）个汉字并显示出来，下一行余下的汉字则再分配到它的下面一行，以此类推。程序中140——180语句保证截取的最后一个汉字是完整的。

注：1. 对用WORDSTAR编辑的文本文件，要求每行有一个硬回车。（即每一行敲完，人为按一下回车键）。

2. 本程序已在IBM PC机上调试通过。

中文阅读程序清单：

```
]LIST
10 REM 文件阅读程序
20 INPUT “请输入您要阅读的文件名：”,D$
30 INPUT“请输入显示方式：D（大汉字），X（标准字）：”，S$
40 IF S$ = “D” OR S$ = “d” THEN H=40,SCREEN 1,0,0,COLOR 0,1,1,GOTO 60
50 IF S$ = “X” OR S$ = “x” THEN H=80,WIDTH 80 ELSE PRINT “输入错 ！ ”,GOTO 30
60 INPUT “请输入控制方式：D（定时),S（手控）：”,S1$
70 OPEN D$ FOR INPUT AS#1,OPEN “SCRN,” FOR OUTPUT AS#2
80 J=0,C$ = “”,CLS
90 IF EOF(1) THEN GOSUB 250,GOTO 210 ELSE LINE INPUT #1,A$
100 IF LEFT$(A$,1)=CHR$(138) THEN A$=MID$(A$,2)
110 IF A$<>“” AND H=40 THEN I=1,GOSUB 280
120 A$=C$+A$
130 IF LEN(A$)<=H THEN C$ = “”,PRINT #2,A$;,GOSUB 240,GOTO 90
140 Q=0,IF ASC(MID$(A$,H,1))<128 THEN Q1=H,GOTO 190
150 FOR I=1 TO H
160 IF ASC(MID$(A$,I,1))<128 OR ASC(MID$(A$,I,1))=138 THEN Q=Q+1
170 NEXT I
180 IF Q/2=INT(Q/2) THEN Q1=H ELSE Q1=H-1
190 PRINT #2,MID$(A$,1,Q1);,C$=MID$(A$,1+Q1),GOSUB 240
200 IF LEN(C$)>H THEN A$=C$,GOTO 140 ELSE GOTO 90
210 CLOSE,INPUT“重复否 ”; X$,IF X$ = “Y” OR X$ = “y” THEN 70
220 INPUT“还阅读别的文件吗 ”;X$,IF X$ = “Y” OR X$ = “y” THEN 20
230 WIDTH 80,END
240 J=J+1,IF J<10 THEN PRINT,RETURN
250 J=0,IF S1$ = “S” OR S1$ = “s” THEN INPUT “”,S$ ELSE GOSUB 270
260 CLS,RETURN
270 FOR U=1 TO 19000,NEXT U,RETURN
280 IF MID$(A$,I,1)=CHR$(32) THEN I=I+1,GOTO 280
290 A$=MID$(A$,I),I=1
300 IF ASC(RIGHT$(A$,I))=32 THEN I=I+1,GOTO 300
310 A$=MID$(A$,1,LEN(A$)-I+1),RETURN
```

机器语言子程序的连结方式

湖南省株州市南方公司职工大学　熊锡义

在BASIC程序里常常要用到机器语言子程序，根据机器语言子程序的长短，可以选用不同的方式把它与BASIC程序连结起来，常用的方法有三种：

1) POKE的方式：

如果机器语言子程序不长，可以用POKE的方式把子程序放在第三页（768～1023），除去1008以后的16个位址外，共有240个位址可供使用。使用POKE的时候，必须将位址及机器语言指令的16进制数码转换成10进制，在机器语言子程序的末尾应有返回指令RTS（$60，十进制96），BASIC语言的CALL指令应指向机器语言子程序的首位址。程序一是一个编辑型LIST程序，它的功能是用LIST指令列印时，所有的空格都不会显示，显示宽度为33格。如果你用游标进行编辑（ESC　I、J、K、L）以后，也不会把多余的空格列出。

使用POKE的时候，必须将数据和代码转换成16进制，这对于较长的程序是不方便，能否将16进制的位址和机器码直接送进APPLE里呢？我们可以用BASIC字串的形式进行。程序一

```
10 FOR X=0 TO 7:READ A:POKE 768+X, A:NEXT
20 DATA 201, 160, 208, 1, 96, 76, 240, 253
30 POKE 54, 0: POKE 55, 3: CALL 1002: POKE 33, 33
```

2) 以字符串的方式读入第2页

程序二

```
10 Y$="300:20 1B FD C9 83 D0 06 20 66 D5 20 D0 03 60 N 38:00 03 N A851G N D823G"
20 FOR I=1 TO LEN (Y$): POKE 511+I, ASC (MID $ (Y$, I, 1)) +128:NEXT
30 CALL -144
```

现在用程序二来说明，字符串变量Y$里存放着监控程序里可以完成的四条立即指令，第一，把$20、$1B…依次存入$300、$301…位址。第二，把$00、$03分别存入$38、$39。第三，执行$A851开始的机器语言程序。第四，执行$D823开始的程序。接着，将字串逐个分解以后，以ASCII码的形式存入第二页，这是一个由512位址开始的键盘缓冲区，请注意，由于监控程序中使用的是负ASCII码，因而ASC函数值加上了128。CALL-144是呼叫监控程序读人键盘缓冲区的资料并执行。Y$中字符串的字符数应小于256，否则，第二页将容纳不下。

程序二可以用来对你的程序加密。我们知道，用POKE1010，102：POKE1011，213：CALL-1169，可以防止别人用CTRL-RESET键来中断程序，再把你的程序LIST出来。但无法防止CTRL-C键来中断，尤其是中文程序。程序二是用栏截CTRL-C的方法来防止中断，当程序发现按下的是CTRL-C（$83）即转向$D566，从而继续执行程序。仿此，我们可以栏截其他任何键（如CATALOG等）。

3) 位址重整法：

如果机器语言子程序的长度超过一页（256），就不能用前面两种方法，我们可以把它放在高位址上，如 $6000～$BFFF（10进制24576～49151）。常用的方法是先用CALL-151进人监控状态，键人机器语言子程序，再将BASIC程序和机器语言子程序分别用SAVE和BSAVE存人磁盘，使用时又得分别调人内存，这样未免太费时费事了。使用调整位址的方法，可以将BASIC程序和机器语言子程序一次存入或调出。

在零页中，有一对指针（$AF，$BO）它是指向BASIC程序的最高位址的再高一位址，当使用SAVE指令存放BASIC程序时，就将$800至该指针所指的位址之间的资料全部存人磁盘。如果我们找出（$AF，$BO）所指之值，然后从该位址之后约10个字节起，键人机器语言

二元一次方程组练习题程序

周慕贤

这个程序运行时，计算机能随机产生出二元一次方程组供学生练习，并能判断学生的解答是否正确，同时根据学生解答的情况给予评分。出题时每次以5个方程组为一套，每题20分，共100分。每题最多给学生四次回答的机会，如果第一次就答对，则该题给20分，并加入总分中；如果第二次才答对，则该题给15分……（每多做一次则该题得分减少5分）。如果第四次还答错，则该题给0分，并显示出正确的答案。

本程序由30句随机产生X值和Y值（取值范围是±0.5、±1、±1.5……±10），由35句随机产生两个方程的系数A（1）、B（1）、A（2）、B（2）、（取值范围是±1、±2……±10）（A（K）只取正值）。虽然系数及其解都比较简单，但从理论上说，所产生的方程组至少要几千万次才可能出现重复，所以可供学生多次练习。而且实际运行证明，一般学生要得100分还不很容易呢！不信，请你也试试看。

程序中40－50句的作用是打印出两个二元一次方程式以组成方程组，并使其写法与习惯写法相同。60～90句用以判断学生的解答并根据解答的情况计算得分。当做完5题之后，如果还要再做，则只要按下Y键，计算机就会重新出一套题目供你练习。如果认为题目太难，则只要把30句改为

30　X＝INT（－21＊RND（1）＋11）：Y＝INT（－21＊RND（1）＋11）：IF　X＊Y＝0　THEN　30

就可以了。（本程序在APPLE Ⅱ上通过）

程序清单：

```
10 S = 0:M = 1: PRINT "二元一次方程组练习题"
20  PRINT "(";M;")";:N = 1
30 X =  INT ( - 41 *  RND (1) + 21) / 2:Y =
INT ( - 41 *  RND (1) + 21) / 2: IF X * Y = 0
 THEN 30
35  FOR K = 1 TO 2 A(K) =  INT (10 *  RND (1)
 + 1):B(K) =  INT ( - 21 *  RND (1) + 11): NE
XT K: IF B(1) * B(2) = 0 OR A(1) * B(2) = A(2
) * B(1) THEN 35
40  FOR K = 1 TO 2: PRINT " ": PRINT "   ";:
IF A(K) > 1 THEN  PRINT A(K);
42  PRINT "X";: IF B(K) > 0 THEN  PRINT "+";
44  IF B(K) = 1 THEN 50
46  IF B(K) =  - 1 THEN  PRINT "-";: GOTO 50
48  PRINT B(K);
50  PRINT "Y=";A(K) + X + B(K) * Y;: NEXT K:
PRINT "    的解是"
60  INPUT "X=";XX: INPUT "Y=";YY: IF X = XX A
ND Y = YY THEN S = S + 25 - 5 * N: PRINT "
对!您共得了";S;"分": GOTO 90
70  PRINT " 第";N;"次错了":N = N + 1: IF N =
5 THEN  PRINT "答案是X=";X: PRINT "     Y=";Y,
"你仍是";S;"分": GOTO 90
80  GOTO 60
90  PRINT "---------------":M = M + 1: IF M <
6 THEN 20
100  PRINT "还要再做一次吗?如果要,请按下(Y)键
 如果不要请按下任何一个键";: GET Y$: IF Y$ =
 "Y" THEN 10
110  END
```

子程序，接着让（$AF，$BO）指向机器语言子程序的最高位址的再高一位址，最后用SAVE将BASIC程序和机器语言子程序一次存入磁盘。这样就省去了频繁启动磁盘的麻烦，节省了程序的运行时间。

利用微机的自动计时功能建立微机工作数据库

武汉汽车配件厂科研所　许传国

微机的开工使用情况是反映计算机使用部门工作成效的一项重要数据。但是有许多计算机室（站）往往还是使用人工方法记录计算机的开机时间、关机时间，手工统计计算机的工作情况，这对于使用高度自动化的工具来进行科学管理的部门来说，不能不是一个很大的遗憾。IBM—PC／XT及其兼容机具有自动计时功能。每次开机时，只要把当天日期和当时时间输入，微机的自动计时系统就能以此为依据计算其后每一工作时刻的日期和时间。DBASEⅢ还具有系统日期（DATE（））函数和系统时间（TIME（））函数，能进行日期函数的运算。利用PC机的以上特点，我们在PC微机上建立了一个微机工作数据库。

我们在PC机上建立的微机工作数据库包括二个数据文件。一个数据库文件用来记录微机每次开机、关机的日期、时间及其工作时间，取名为BFTIME，其部分记录及数据库结构如下：

```
USE BFTIME
LIST FOR 开机日期 > CTOD('01/08/87')
Record#  开机日期 开机时间 关机日期 关机时间 工作时间
     34  01/09/87 07:09:50 01/09/87 11:18:39     4.14
     35  01/09/87 14:16:09 01/09/87 16:15:03     1.98
     36  01/10/87 07:02:45 01/10/87 11:18:39     4.26
     37  01/10/87 13:03:42 01/10/87 16:17:34     3.23

A>TYPE BFTIME.TXT
Structure for database : A:bftime.dbf
Number of data records :      37
Date of last update    : 01/10/87
Field  Field name  Type       Width Dec
    1  开机日期    Date           8
    2  开机时间    Character      8
    3  关机日期    Date           8
    4  关机时间    Character      8
    5  工作时间    Numeric        6    2
** Total **                      39
```

另一个数据库文件用来累计微机的工作机时，使用天数及记载始用日期、截止日期和使用率，取名为JSLJ，仅有一条记录，其记录及数据库结构情况如下：

```
. USE JSLJ
. LIST
Record#  始用日期 截止日期        累计机时        使用天数 使用率
      1  12/01/86 01/10/87          163.43              31   5.34

A>TYPE JSLJ.TXT
Structure for database : A:jslj.dbf
Number of data records :        1
Date of last update    : 01/10/87
Field  Field name  Type       Width  Dec
    1  始用日期    Date           8
    2  截止日期    Date           8
    3  累计机时    Numeric       15    2
    4  使用天数    Numeric       15
    5  使用率      Numeric        5    2
** Total **                      52
```

使用方法是：在微机的自举启动程序（AUTOEXEC.BAT）中加入DATE和TIME二条命令。这样，在开机自举后，根据开机时的时间输入系统日期和系统时间。然后使系统进入DBASEⅢ，打入命令DO　BFON［回车］微机就在DBASE状态下执行BFON程序，根据程序执行时的系统日期和系统时间把有关开机日期和开机时间的信息记入BFTIME数据库新记录的有关字段中。在工作完毕，关机以前，使系统再进入DBASEⅢ，执行一个BFOFF的程序（在键盘上打入DO　BFOFF［回车］），程序就计算此次开机的工作时间（以小时为单位），并和此时的系统日期、系统时间信息一起记入BFTIME数据库在开机时建立的新记录的有关字段中。接着，程序把此次关机时的日期作为截止日期记入JSLJ数据库的截止日期字段，把工作时间累计到累计机时字段中去，根据开机情况使使用天数字段累加上适当的天数（0天，一天或几天不等），并根据累计机时，使用天数计算其平均值（使用率），记入使用率字段。在JSLJ数据库记录完毕，DBASE圆点提示符出现后，操作者就可以关机了。

当遇到突然停电或者其他原因而在关机前不能

执行BFOFF程序使数据库中数据不全时，可在事后用DBASEⅢ的编辑命令进行修改，以作补救。

我们在实际使用时，是把DBASEⅢ的装人及BFON的执行都放在系统的自动启动程序中，自动进入DBASEⅢ状态及执行BFON程序。在BFON执行完后自动运行另一个DBASEⅢ下的管理系统的目录程序。这样，只要在关机前执行一次BFOFF程序就行了，十分方便。

当然，在微机工作数据库的管理上还可以搞得更细一些。但是有这二个库作为基础，其他的分类，统计都可迎刃而解，就不多述了。

几点说明：

一. BFOFF. PRG 程序中，第25行，plustw=累计机时后面有7个空格符，程序清单中看不出，输入程序时应键人。因为累计机时字段有15个字节，而“累计机时”四个汉字仅占8个字节，因此“累计机时”后面要补7个空格，以便和该字段宽度相配。同样，第26行，ts=使用天数后也有7个空格；第34，35行的累计机时，使用天数后面也有7个空格，输程序时不要忘了键人，否则程序执行时会出错。

二. BFOFF. PRG程序调用一个过程TTOS. PRG，因为数据库要根据开机时间、关机时间计算工作时间。该过程把开机时间，关机时间的时、分、秒（以0时0分0秒为基准）统统化成秒，以简化计算。

本数据库在IBM PC／XT的兼容机（北方电脑公司的BF PC／XT PLUS机）上实现。

程序清单如下。

```
. TYPE BFON.PRG
SET TALK OFF
store date() to db
store time() to tb
store ctod('  /  /  ') to de
store "" to te
store 0 to tw
use C:bftime
append blank
replace 开机日期 WITH DB
replace 开机时间 WITH TB
replace 关机日期 WITH DE
replace 关机时间 WITH TE
replace 工作时间 WITH TW
clear all
do a:menu.prg
```

```
. TYPE BFOFF.PRG
set procedure to ttos
SET TALK OFF
USE C:BFTIME
GOTO BOTTOM
DB=开机日期
TB=开机时间
store date() to de
store time() to te
store 0 to tw
d=de-db
store tb to o
store 0 to second
do ttos with o
store second to t1
store te to o
do ttos with o
store second to t2
close procedure
 tw=(d*24*3600+t2-t1)/3600
replace 关机日期 with de
replace 关机时间 with te
replace 工作时间 with tw
use C:JSLJ
dj=截止日期
plustw=累计机时
ts=使用天数
sr=使用率
plustw=plustw+tw
if dj<>de
  ts=ts+d+1
endif
sr=plustw/ts
replace 截止日期 with de
replace 累计机时        with plustw
replace 使用天数        with ts
replace 使用率 with sr
clear all
close procedure
clear
@ 3,26 say "@@@@ 再        见 @@@@"
cancel
```

```
. TYPE TTOS.PRG
procedure ttos
parameters o
l=len(o)
store 0 to point1
store 0 to point2
store 0 to second
do while point1<l
store point1+1 to point1
if substr(o,point1,1) <> ':'
loop
endif
if point2=0
second=val(substr(o,1,point1-1))*
*60*60+second
store point1+1 to point3
endif
point2=point2+1
if point2<2
loop
```

APPLESOFT语言程序复盖技术初探

福建宁德师专计算机室 陈 昕

在开发和应用APPLE—Ⅱ微机中，由于内存只有48K，因此，在编制较大型的BASIC程序时，可能由于程序过长而无法全部装入内存，即使能装入，也可能因为程序占用过多的内存，而没有足够的存贮空间用来存放运行中的变量值，而使整个程序无法正常运行。解决的办法就是用所谓“模块化”程序设计方法，把整个程序分为几个较小的功能模块分别设计，使每个模块均能在48K内存中正常运行，然后分别以模块为单位存盘。运行时先调人某个功能模块，待其完成特定功能后，再调入后继模块，将前一模块复盖，并继续执行后继的模块，如此下去，直至实现整个程序目标。这就是所谓的程序复盖技术。

在APPLESOFT（含DOS3.3）状态下，如何实现程序复盖技术呢？以下对此作一介绍。

1.模块间的全复盖方法

假定有两个模块，其存盘名分别为M—1和M—2，现要用M—2将M—1全部复盖；这只须在模块M—1的最后加上一行：

```
1000 PRINT CHR $ (4) ; "RUN M—2"
```

即可。但复盖后，M—1中的所有变量或数组值均被抹去，模块M—2无法使用这些值。

如果要使复盖后，M—1中的变量或数组值为模块M—2的程序所使用，可用CHAIN程序。其方法是：先把DOS3.3主磁盘上一个名为CHAIN的机器语言程序，拷贝到已存有M—1和M—2模块的磁盘上（拷贝方法DOS手册上均有介绍）；其次，在M—1模块的最后加上两行：

```
1000 PRINT CHR $ (4) ; "BLOAD CHAIN, A520"
1001 CALL 520 "M—2"
```

这样，当M—1运行完毕后，就会自动调人M—2将其复盖，并把M—1中的变量或数组值传递给M—2使用。必须注意：1001句中520与"M—2"之间不许有任何字符，包括空格符。

2.模块间的部分复盖方法

我们知道BASIC程序是从LOMEN所指定的存贮单元开始装入内存的，因此，可用拨动LOMEN指针的方法实现模块的部分复盖。在复盖前，先把LOMEN指针拨向不被复盖部分的最后一个语句的末尾，然后装人另一模块并运行之，而第二个模块的第一条语句又把LOMEN指针拨向$801。这样就实现了模块的部分（事实上，是后半部）复盖。下面举个例子具体说明做法：

假定软盘上已存有CHAIN程序，并存有：

模块A：

```
10 DIM A(11)
20 PRINT "1 CALCULATION"
30 PRINT "2 EXIT"
40 INPUT "WHICH WOULD YOU LIKE?",N
50 IF N<0 OR N>2 THEN40
60 ON N GOTO 80,70
70 HOME:END
80 FOR I=1 TO 11
90 INPUT A(I)
100 NEXT
```

模块B：

```
200 S=0
210 FOR I=1 TO 11
220 S=S+A(I)
230 NEXT
240 S=S/11
250 PRINT "S=";S
260 GOTO 20
```

现在，要使A模块只从80行以后被B模块复盖，并把A模块的变量和数组值传递给B模块。这

```
endif
second=second+val(substr(
(o,point3,point1-1))*60
second=second+val(substr(
(o,point1+1,l))
exit
enddo
return
```

怎样提高APPLE—Ⅱ机的计算精度

西北工业大学　赵政文

众所周知，APPLE机的有效位数只有九位，对于普通的数学计算，精度是满足的。但是在实际应用中，常常会发现用户所要求的有效数字超过九位，特别是在财务结算，经济预测等方面的应用中，有效数位要求在12位以上，如果直接计算，显然要丢失有效数位。因此需要用程序来实现这一问题的求解。这实际中，我们采用类似于汇编语言中的高低位数方法，将数字分成两段，具体分法是将数中低位的9位做为一个数段，高于9位的做为另一个数段，例如：123003456789可分为：123和003456789。对于这样的数，采用高低数段分别进行计算，然后以字符串形式输出的方法，既保证了有效数位，又能进行各种计算。但是由于机器中处理数时，会将一个数中第一个有效数字前的“0”舍去，如果直接进行字符串的形式输出，则有可能改变其值。在前面的例子中，如果直接输出字符串，则会出现下面的情况：设A＝1 2 3 0 0 3 4 5 6 7 8 9 按前面的分段法，A1 ＝1 2 3，A2 ＝0 0 3 4 5 6 7 8 9，那么字符串形式输出则会是：S T R $（A1）+S T R $（A 2）＝1 2 3 3 4 5 6 7 8 9 使其结果值不等于1 2 3 0 0 3 4 5 6 7 8 9 。另外，在计算中还有高段和低段数的正负号一致问题。在实际应用中，解决这类问题我们考虑了如下几个方面的问题：

1. 保留小数点后的有效数字，即给原数乘以10^m。（m为小数点后有效数位）

2. 设置一个加（减）法子程序段，对于任意两个数，进行高低数段的加（减）运算，并使其结

只须在A模块的末尾加上三个语句行：

```
110 PRINT CHR$(4);"BLOAD CHAIN,A520"
120 POKE 103,131:POKE 104,8
130 CALL 520"B"
```

而在B模块的最前面加上一个语句：

```
190 POKE 103,1,POKE 104,8
```

这样，当模块A运行结束后，即可将B模块调入内存，将A模块80行以后的部分复盖，并把变量及数组值传递给B模块。如果不用做变量传递的话，则只要将上述110－130行改为：

```
120 POKE 103,131,POKE 104,8
130 PRINT CHR$(4);"RUN B"
```

即可。

在这里120行POKE语句中的两个数131和8是这样选取的：首先调入A模块，删去80行以后的语句，然后，键入命令：

```
] PRINT PEEK(105)-5,PEEK(106)
```

得到：129　　8

将其分别化为16进制数 $81（即129）和 $08（即8）组成一个内存地址：$0881；再键入命令：] CALL—151，进入监控状态：*，然后打入上面的地址码 $0881，即可见到：

```
*0881—80 [回车]
*
00 00 00 0A 43 81
```

记下三个连续0字节的中间那个字节的地址：$0883。最后，将此地址的低两位 $83和高两位 $08，分别化为十进制数131和8，这就求得POKE语句中的两个数。注意：是将地址低两位十进制数POKE到103单元，高两位十进制数POKE到104单元。

为了使部分复盖后的程序能正常运行，在编程序时，要特别注意行号的选取：其要求是复盖模块的最小行号，不能小于未被复盖的语句的行号。如上例中，B模块的行号必须从大于70的正整数开始取。

上面初步探讨了程序复盖技术，一般地说，如果模块的独立性较强，则用第一种方法较为简捷和方便。第二种方法的优点在于：能将被复盖模块的部分程序段，为复盖模块所利用。这样，在编程序时，只须把各模块都要用到的程序段或子程序，编在第一个被调入内存的模块内，通过部分复盖，即可使各模块共享这些程序段或子程序；这不仅节省了磁盘存贮空间，而且，能缩短模块调入时间和减少编程中的重复劳动。

果数的高低数段正负号一致。而且使其低数段化为小数形式，即使低数段除以10^m。

3. 设置一个将高低数段化为字符串形式并使其相并后输出。主要解决低数段第一个有效数字前的若干个“0”不被丢失。

程序清单如下：

1. 加法子程序清单。200到390

```
200 C1 = 0: C2 = 0
205 C1 = A1 + B1
207 IF ABS (A2 + B2)
    < 1000000000 THEN
    C2 = A2 + B2: GOTO
     245
210 IF (A2 + B2) > 100
     0000000 THEN 215
212 IF (A2 + B2) < -10
    00000000 THEN 225
215 C1 = C1 + 1: A2 =
    A2 - 500000000: B2
    = B2 - 500000000
220 C2 = A2 + B2: GOTO
    245
225 C1 = C1 - 1
230 A2 = A2 + 5000000
    00: B2 = B2 + 500
    000000
235 C2 = A2 + B2
245 IF C1 > 0 AND C2
    < 0 THEN 255
250 IF C1 < 0 AND C2
    > u THEN 265
253 RETURN
255 C1 = C1 - 1: C2 =
    C2 + 500000000: C2
    = C2 + 500000000
260 RETURN
265 C1 = C1 + 1: C2 =
    C2 - 500000000: C2
    = C2 - 500000000
390  RETURN
```

2. 字符串化子程序清单。600到720

```
600 IF HW = 0 AND LW =
    0 THEN WA$ = "" :
    RETURN
605 IF HW = 0 AND LW
    < > 0 THEN WA$ =
    STR$ (LW): RETURN
607 IF HW < > 0 AND LW
    = 0 THEN WA$ = STR$
    (HW) + "0000000":
    RETURN
608 IF ABS (LW) > 1000
    000 THEN WA$ = STR$
    (HW) + STR$ ( ABS (
    LW)): RETURN
610 IF ABS (LW) < 1000
    000 AND ABS (LW) >=
    100000 THEN L = 1:
    GOTO 700
615 IF ABS (LW) < 1000
    00 AND ABS (LW) >=
    10000 THEN L = 2:
    GOTO 700
620 IF ABS (LW) < 10000
    AND ABS (LW) >= 1000
    THEN L = 3: GOTO 700
625 IF ABS (LW) < 1000
    AND ABS (LW) >= 100
    THEN L = 4: GOTO 700
630 IF ABS (LW) < 100
    AND ABS (LW) >= 10
    THEN L = 5: GOTO 700
635 IF ABS (LW) < 10 AND
    ABS (LW) >= 1 THEN
    L = 6: GOTO 700
637 IF ABS (LW) < 1 AND
    ABS (LW) > 0 THEN
    L = 7
700 WA$ = STR$ (HW)
705 FOR I = 1 TO L
710 WA$ = WA$ +"0"
715 NEXT I
720 WA$ = WA$ + STR$
    ( ABS (LW)): RETURN
```

如何在DOS3.3上使用35条以上磁轨

广州市四十六中　刘志仁

APPLE Ⅱ微机是我国应用较多的机种。在APPLE Ⅱ机上使用的是 5.25 英寸的单面单密度软磁盘。现在一般均使用DOS3.3磁盘操作系统，APPLE公司在推出DOS3.3时（1980年 8月25日），规定了此系统支持下的磁盘是使用35个磁轨，每磁轨16个扇区，每个扇区最多可存放256个字节。

但是，使用磁盘时，第0，1，2三条轨道保留给DOS使用，17轨道被用来存放磁盘文件的目录，只剩下31条磁轨留给用户使用。这样，人们在使用时，不时会觉得磁盘的容量太少。因此，有些人便舍去DOS，让磁盘的第1，2磁轨释放出来，以增加磁盘的容量。但是，这样磁盘就不保留DOS，使用起来总觉不便。

是否有办法增加磁盘的容量呢？到了今天，由于磁盘质量的提高，以及磁盘驱动器为使其能为各种微机兼容，质量也不断提高，性能也有所改进，使得在APPLE Ⅱ上的磁盘驱动器的读／写磁头能够读写到35磁轨以外的地方。而且，APPLE公司在推出DOS3.3时，也为以后使用35个以上的磁轨留有余地，使到我们在使用多于35条磁轨时，不致于要对DOS系统作太多的改动。

既然它能够读／写35条以外的轨道，我们不用，其实是一种浪费。哪么，怎样去使用35条以上的轨道呢？其方法很简单。首先，将标准的DOS3.3放入内存，然后将$BEFE（48894）单元的值由$23改为$30，再执行INIT命令，结束后，在$11（17）轨道0扇区的卷目录表（VTOC）中的$C0以后的字节填上$FF，以表示未使用。这样处理后，我们就可以使用35条以上的轨道了。

为了使读者更方便地将磁盘格式化成多轨道的磁盘，本人编了一个程序，只要将程序运行一次，便能将磁盘格式化成可以使用35条以上磁轨的磁盘。程序不长，可以直接从键盘敲入，也可以将程序存入磁盘，待需要时方便地调用。注意的是，在运行本程序之前，先要存入DOS3.3，及一个BASIC的HELLO程序。

这里还有一个问题，就是用户最多可以使用多少个磁轨。从磁盘的卷目录表看，最多可使用是50个磁轨，从$C0～$FF可以记录15条磁轨的使用情况，加上原来的35条轨道，便是50条轨道了。这是DOS3.3留出的空间，但很遗憾，目前，还未发现有磁盘驱动器可以读／写这么多的磁轨，至于到底能使用多少磁轨呢？这没有一个确定的数字，从实践看，不同的磁盘驱动器其数值不一样，一般都能使用40条磁轨。总之，无论使用什么型号的驱动器，都能使磁盘的容量增加，并且，经以上程序处理后，使用时，不会对DOS系统产生不良影响，用户可照常使用。

```
*2000.215B

2000-  20 58 FC A0 00 B9 00 21
2008-  C8 20 F0 FD C0 27 D0 F5
2010-  20 8E FD A2 28 20 4A F9
2018-  A0 00 B9 40 21 C8 20 F0
2020-  FD C0 1D D0 F5 A2 28 20
2028-  8E FD 20 4A F9 A0 00 B9
2030-  27 21 C8 20 F0 FD C0 0C
2038-  D0 F5 20 1B FD 20 F0 FD
2040-  8D 6A AA 20 1B FD AA E0
2048-  8D D0 EF 20 8E FD A0 00
2050-  B9 33 21 C8 20 F0 FD C0
2058-  0D D0 F5 20 1B FD 20 F0
2060-  FD 8D 68 AA 8D 0C 30 20
2068-  1B FD AA E0 8D D0 EC 20
2070-  8E FD A9 00 8D 5F AA 20
2078-  95 A0 A9 C8 8D 75 AA A9
2080-  C5 8D 76 AA A9 CC 8D 77
2088-  AA 8D 78 AA A9 CF 8D 79
2090-  AA A9 70 8D 65 AA A9 32
2098-  8D FE BE 20 4F A5 A9 FF
20A0-  A2 00 9D 7F B4 9D 80 B4
20A8-  E8 E8 E8 E8 E0 A0 D0 F2
20B0-  A9 00 A9 32 8D EF B3 A0
20B8-  00 B9 EE 20 99 0D 30 C8
20C0-  C0 12 D0 F5 A9 01 8D 0A
20C8-  30 A9 60 8D 0B 30 A9 30
```

在LASER310上使用ON语句

安徽省蚌埠印染厂仪表室　　施国范

使用ON—GOTO语句可实现多分支转移，而在LASER310机的BASIC手册中却没有这个语句。直接从键盘输入ON语句，电脑将不予响应。虽然可采用多个IF--THEN语句来实现多分支转移，但是却使程序复杂化了。然而在LASER310的驻机解释程序中，却有一段ON语句的工作子程序。只要采用本文所介绍的方法，即可在LASER310机上使用ON—GOTO语句。

LASER310不能直接使用ON语句的主要原因，是在驻机解释程序的保留字代号表中没有ON的代号，当程序运行时就无法找到ON子程序的入口。因此我们只要设法找到ON的代号，并把它写入到程序中ON的位置上，那么电脑就可以执行ON功能了。

经分析，得到ON的代号为A1H（十进制为161）。另一个问题是如何找到ON所在行的首地址？这可以使用程序1而方便地找到。在BASIC应用程序的后面接着输入这一程序，RUN5010运行后，屏幕上显示“L=？”，这时输入ON所在行的行号，计算机即能显示出该行的首地址和下一行的首地址。

具体使用方法如下，假如要输入的应用程序如程序2。在输入过程中先用一个符号（如&）来代替程序中的“ON”，用意是给后面重新写入ON的代号而保留一个内存空间。待程序全部输入完后（包括上述程序1），先运行程序1（即RUN5010），输入ON所在行号L=20，得到20行的首址为31473。根据BASIC程序在内存中的贮存规律，将该地址加4，即得到写入ON代号的位置为31477。用BREAK键退出程序1。再键入POKE31477，161命令，ON的代号就写到既定位置上了。至此程序2即可投入正常运行。这段程序的用意是当输入N=1，2，3，4时，分别打印出1至4个星号。

值得注意的是在输入过程中，只能用一个ASCII符号来代替ON，并且最好用程序中所不使用的符号，以免混淆或出错。本方法有一个不足之处，就是当LIST时，屏幕上将不显示ON所在行中自ON起以后的字符。

程序1

```
5010 INPUT "L=";L:A0=31465
5020 AN=PEEK(A0)+256*PEEK(A0+1)
5030 LT=PEEK(A0+2)+256*PEEK(A0+3)
5040 IF AN>32767 THEN A1=AN-65536 ELSE
     A1=AN
5050 IF LT=L THEN 5070
5060 A0=A1 : GOTO 5020
5070 L$=STR$(L)
5080 PRINT "A("+L$+")=";A0,"A(NEXT)=";A1
5090 GOTO 5010
```

程序2

```
10  INPUT N
20  ON N GOTO 100,200,300,400
100 PRINT "*":GOTO 10
200 PRINT "**":GOTO 10
300 PRINT "***":GOTO 10
400 PRINT "****":GOTO 10
```

```
20D0- A0 0A 20 D9 03 A9 C5 20
20D8- F0 FD A9 CE 20 F0 FD A9
20E0- C4 20 F0 FD 20 8E FD 20
20E8- 3A FF 20 B3 FE 00 00 11
20F0- 00 1B 30 BB B3 00 00 02
20F8- 00 00 60 01 00 01 EF D8
2100- A0 C7 D5 C1 CE C7 DA C8
2108- CF D5 A0 A0 C6 CF D2 D4
2110- D9 AD D3 C9 D8 D4 C8 A0
2118- A0 CD C9 C4 C4 CC C5 A0
2120- A0 D3 C3 C8 CF CF CC A0
2128- A0 A0 A0 A0 D3 CC CF D4
2130- A0 A3 BA A0 A0 A0 A0 A0
2138- C4 D2 C9 D6 C5 A0 A3 BA
2140- A0 A0 A0 A0 A0 A0 A0 A0
2148- 4D 4F 52 45 A0 A0 54 52
2150- 41 43 4B A0 A0 46 4F 52
2158- 4D 41 54 45
```

dBASE ⅢPLUS数据管理标准命令语法

彭荣昌 译 陈锦荣 校

（接上期）

在全屏幕方式中不跳或跳到下一字段。

SET CONSOLE ON/off

向屏幕发送或不发送所有的输出结果。

SET DATE AMERICA/ANSI/BRITISH/ITALIAN/FRENCH/GERMAN

决定日期表示式的格式，DATE（日期）通常是放在AMERICAN上。

SET DEBUG on/OFF

发送或不发送SET ECHO的输出到打印机。

SET DECIMALS TO <expN>

使十进制的最小数显示在某些操作和功能的结果中。

SET DEFAULT TO <drive>

为方便查找档案指定缺席驱动器。

SET DELETED on/OFF

隐藏或处理标有删去符号的记录。

SET DELIMITERS on/OFF

用或不用指定的输入定界符。

SET DELIMITERS TO [<character string>][DEFAULT]

为全屏幕字段和变数显示指定分界符。

SET DEVICE TO SCREEN/print

把@……SAY命令的结果发送到屏幕或打印机。

SET DOHISTORY on/OFF

命令档案记录在HISTORY中是否确定。

SET ECHO on/OFF

回送或不回送命令行到屏幕上或打印机上。

SET ESCAPE ON/off

当ESC键被按下时，停止或继续命令档案的执行。

SET EXACT on/OFF

在字符比较中要求或不要求恰当的匹配。

SET FIELDS on/OFF

重视或忽视指定的字段目录。

SET FIELDS TO [<field list>ALL]

确定那些字段可存取。

SET FILTER TO [FILE<.qry filename>/?][<condition>]

使一个数据库档案呈现出似乎它只包含那些适合指定条件的记录。

SET FIXED ON/OFF

固定或不固定要显示的十进制数目。

SET FORMAT TO [<.fmt filename>/?]

打开一个格式化档案以便数据进入。

SET FUNCTION<exp>TO<expC>

设置功能键数值。

SET HEADING ON/off

字段名称显示或不显示在LIST或DISPLAY的字段上。

SET HELP ON/off

当发生错误时，为了帮助使用者给提示或不提示。

SET HISTORY ON/off

把命令的HISTORY转至开或关。

SET HISTORY TO <expN>

指定执行过的命令数字存储在HISTORY（历史）档案中，存储命令的缺席码是20。

SET INDEX TO [<.ndx file list>/?]

打开已取名的索引档案。

SET INTENSITY ON/off

使用或不使用加强的显示（器）。

SET MARGIN TO <expN>

设定打印机的左边缘。

SET MEMOWIDTH TO <expN>

调整存储字段输出的列宽，缺席值为50。

SET MENUS ON/off

在全屏幕命令期间，显示或不显示菜单。

SET MESSAGE TO <cstring>

在屏幕的底行上显示使用者定义的信息。

SET ORDER TO [<expN>]

当控制档案没有关闭和重新打开档案时，建立任一个打开的索引档案。

SET PATH TO [<path list>]

指定一个路径目录以便查找档案。

SET PRINT on/OFF

发送或不发送输出结果到打印机。

SET PRINTER TO <DOS device>

把输出结果引至指定的设备。

SET PROCEDURE TO [<procedure filename>]

打开已取名的程序档案。

SET RELATION TO [<key>/RECONO( )/<expN>INTO<alias>]

按照键的表示式去联结两个数据库档案。

SET SAFETY ON/off

当一档案即将被写过头时给出提示或不给提示。

SET STATUS ON/off

根据点提示符，显示或不显示状态行。

SET STEP on/OFF

在每一个命令被处理后，暂停或不暂停程序的执行。

SET TALK ON/off

命令执行的结果，发送或不发送到屏幕。

SET TITLE ON/off

当一个新的档案被加到目录上时，对一个目录档案的标题给提示或不给提示。

SET TYPEAHEAD TO <expN>

指定typ—ahead缓冲器的大小。20，是缺席大小。

SET UNIQUE on/OFF

具有相同键的第一记录或所有的记录出现在一个索引档案中。

SET VIEW TO <.vue filename>/?

打开一个.vue档案。

SKIP<expN>

以它当前的位置为基准，使当前的记录符向前或向后移动。

SORT TO <new filename> ON <field>[/A][/C][/D][,<field2>[/A][/C][/D]……][<scope>][WHILE<condition>][FOR<condition>]

按照一个或多个数据字段，创建一个有序的数据库拷贝。

STORE<exp>TO<memvar list>[,<memvar list>]

把一个表示式存储进一个或多个存储变数中。

SUM [<scope>][<exp list>]TO[<memvar list>][WHILE<condition>][FOR<condition>]

计算和显示范围内所指定的数据库记录表示式的总和。

SUSPEND

停止执行当前正在执行命令的档案或程序并显示点提示符. 用RESUME继续执行。

TEXT……ENDTEXT

从一个命令档案中显示一个文本数据块. TEXT必须用ENDTEXT来终止。

TOTAL TO <filename>ON<key>[<scope>][FIELDS<field list>][WHILE<condition>][FOR.condition>]

创建一个简短的包含数的TOTALs 先分档的数据库。

TYPE <filename>[TO PRINT]

显示文本档案的内容。

UPDATE ON <key field>FROM<alias>REPLACE<field>WITH<exp>[<field2>WITH<exp2>……][RANDOM]

允许对一个数据数库档案进行批量修改。

USE [<.dbf filename>/?][INDEX<.ndx file list>][ALIAS<alias>]

指定要使用的数据库档案，使整个操作一直进行到另一个USE命令发出为止。

WAIT[<prompt>][TO<memvar>]

暂停程序处理一直到一个键被按下为止。

ZAP

从现行的数据库档案中消除所有的记录。（续完）。

使用IBM—FORTRAN语言的几点体会

四川省德阳市东方汽轮机厂数控中心　柳影　王健

在IBM——PC/XT微机上使用FORTRAN（V2.0版本）或FORTRAN77（V3.2版本）作科学或工程计算是比较适宜的。鉴於我们在使用FORTRAN语言建立一个小型的CAM系统时遇到的一些问题。同时结合我们的一些经验谈谈有关使用中的问题，为将准备使用IBM——FORTRAN语言的同志避免一些重复性的劳动。

一、构造个人子程序库

在编译FORTRAN程序时，如果你经常要调用的标准运算子程序或标准的输入输出子程序。均可将其叠加在编译系统的FORTRAN.LIB文件上，作成类似于FORTRAN语言中的标准函数。这样可提高程序的编辑、编译及可读性。

具体做法是：假如你所要建的个人库文件名为×××.FOR，先对其进行FOR1及FOR2（或PAS2）编译后，生成其目标代码文件×××.OBJ。然后键入：LIB［回车］这时屏幕上显示：

```
IBM Personal Computer Library Manager
Version 1.00
<c> Copyright IBM Corp 1984
<c> Copyright Microsoft Corp 1984
Library name：FORTRAN        [回车]
Operations：    +×××        [回车]
List file：                  [回车]
Output  name：  FORTRAN      [回车]
```

这样就把×××.FOR做成了类似FORTRAN库中的标准函数了，构造个人子程序库的数目是没有限制的，在库操作中一次叠加允许的最大数目是八个。若你有认为不适宜的个人子程序库，你只要在库操作中用一×××，就可从FORTRAN.LIB中把它删除掉。

若是你的 FORTRAN 编译系统盘上没有LIB.EXE文件，你可用 MSDOS上的LIB.EXE来建立你的个人子程序库。在编主程序时只要用调用子程序的方法就行了，但要注意必须保证其形参相对应。

二、数据文件的建立

在FORTRAN程序运行时，可以使用由键盘输入数据，也可采用DATA或DATA BLOCK语句。但若是一个程序需要大量的数据，又要进行反复的试算，那末上述方法就有许多不便了。例如键错一个数据，就要重新运行这个程序，要修改一个数据，程序就要重新编译。较好的方法是建立数据文件。具体的做法是使用EDLIN命令：

```
C>EDLIN  DATA1.TXT       [回车]
   New  file
   * I                   [回车]
   1  *
```

建好数据文件后，只要在主程序中用OPEN语句打开文件，再用READ语句读入就行了，但要注意数据文件的格式和顺序必须保持与READ读入的格式和顺序一致。例如：下面程序中，程序先从数据文件DATA1.TXT中读入原始数据进行计算后并把计算结果做成数据文件DATA2.TXT存在盘上以便今后的查询，修改及需要时的再次读入。

```
   DOUBLE PRECISION A, B, C, D
   DIMENSION A (50, 10), B (50, 1
0), C (50, 10), D (50, 10)
   OPEN (2, FILE= 'DATA1.TXT')
   READ (2, *) A, B, C
   .  .


   .  .
   WRITE (*, *) D
   OPEN (3, FILE=' DATA2. TXT, S
TATUS =' NEW' )
   WRITE (3, *)
   CLOSE (3)
   CLOSE (2)
```

```
      .     .
   STOP
   END
```

三、关于数组使用时的问题

我们在扩充到640K的IBM——PC／XT微机上调试程序时发现，其总存储量是分成十段的，即64K一段。一般采用DIMENSION语句格式定义的数组，都是放在同一64K的存储单元中，虽然在主机上增加了内存板，但机器中内存并不增加。当占有内存空间超过64K的数组，在编译时计算机就会产生出错信息：

例如：FOR1编译

```
DOUBLE PRECISION A,B,C,D
DIMENSION A(100,10,3),B(100,10,3  ),C(100,10,3),D(100,10,3)
      .

      .
STOP
END
```

编译后就会出现如下的错误信息：

```
*****Error 808——data exceeds max
pass one    1 Errors Detected
20 Source Lines
```

解决问题的方法是COMMON语句开辟有名公用区来实现转移，使这种占总内存超出64K的多个大数组分别定义到不同的几段：

例如：FOR1编译

```
DOUBLE PRECISION A,B,C,D
COMMON/C1/A(100,10,3)/C2/B(100,10,3)/C3/(100,10,3)


      .
STOP
END
```

上面这个程序中就实现了A，B，C三个数组分配到不同的段，在编译、连结及运行过程中就不会产生错误了，同时在一个有名公用区中可定义几个不同的数组，其条件是总存储量不超过64K。但要注意FORTRAN语言中的规定：有名公用区的名字不能与同一程序中的数组名相同。

一行程序设计技巧一谈

湖北沙市一中　邱小宁

提高优化程序的能力，是每个计算机爱好者的共同愿望，而做一些“一行程序”的训练则是实现这种愿望的有效途径。实践中我们发现，如果程序中非出现分支不可，则该程序很难用一行完成。现以本刊第二届海洋奖第3题为例，说明一种处理分支的技巧。题目是这样的：有一张足够大的厚度为0.15毫米的纸，把它对折，再对折，就这样一直折下去，问折多少次，其厚度刚好等于或大于珠穆朗玛峰的高度？

本刊今年第一期刊出的我校陈志军同学的程序为：

```
 5  A=0.15
 10 A=A*2:X=X+1:IF A<88
    48000 THEN 10
 20 PRINT  X:END
   RUN
   26
```

青少年朋友们，你能够先不看下文，将以上程序改为一行程序吗？

由于10语句后半部分出现分支，好象三个程序行一个也压缩不了。其实不然，一行程序如下：

```
 5  A=.15:FOR I=0 TO 1:A=2*A:X=X+1:I=A>=8848000:NEXT:PRINT X
  RUN
26
```

读完这个程序后，请读者思考两个问题：

1．除IF语句外还有什么语句具有条件转向功能？

2．带有逻辑表达式的赋值语句有什么特点？

另外，这个题目如果改变算法，是否可编出更简洁的一行程序，也请读者一试。

机型：星河——II（与APPLE——II兼容)。

设计安装利用微机测量直流电压的简易接口机

广州市第七十五中学　伍志峰

一、制作目的与本机特点

为了实现记忆测量和用大数字显示结果的这两种希望，现设计安装了一台简易接口机。接入微机后，按显示方式的不同需要，输入不同的程序，就可以成为课堂演示用的理想清晰的数字电压表或“电压——时间”曲线测绘仪。

本机特点是可以对缓慢地一次性变化的电压作扫描显示，由此得出被测电压的极值。另一特点是比7画式数字电压表和指针式电压表的可见度大。此外，因电路简单，易于自制和进一步改进，所以可进一步提高一般中学对微机的利用率。

二、工作原理与制作调试

本接口机的工作原理，是将被测电压转换为与之成正比的非零电平时间。送入微机的游戏摇杆插口，利用微机的模数转换功能，测得一个函数（0～255）。于是输入适当的程序，再换算出被测电压值，用高分辨作图功能绘出结果。图(一)是原理方框图。图(二)是接口机的电路图。

由R1两端引出电压测试笔。输出端接APPLEⅡ型微机的游戏插座第6脚，地线接第8脚。

本机使用±6V两个电源。电压必须稳定，否则会严重影响读数的稳定性。5G28是高阻抗输入的运算放大器。

制作步骤是：先令被测电压为零，调节微调电阻W，使5G28的7脚电位为零。并用两个固定电阻代替W。使电路工作稳定。再与标准电压表一齐监测。这时对微机输入的程序为：

```
10 X=PDL(0)：PRINT X；" "；
20 FOR I=1 TO 1000：NEXT I
30 GOTO 10
```

记录一组对应于被测电压的函数值，并绘出一条坐标曲线，如图(三)所示。如果所得不是成直线，就要调节R11的大小。调好R11后，取直线上任意两点的数值（U_1，X_1）和（U_2，X_2），及零伏之值X_0，算出线性方程

$$U=KX-B$$

中的斜率$K=\frac{U_2-U_1}{X_2-X_1}$ 和U轴截距$b=KX_0$

代入程序中的换算公式。例如：当零伏时，

$X_0=31$ 当$V_2=2.6$伏时 $X_2=223$

$$则\ K=\frac{2.6-0}{223-31}=0.0135135135$$

$$b=0.418918918$$

把K和b代入附（一）程序第8行号第2语句中的线性方程。把$X_0=31$和$X_2=233$代入附（二）程序第40行号第2语句中。

使用方法是：将两个程序命名后录入磁盘。当要用作数字电压表时，接好接口机，按下“RUN DV—METER”命令，用接口机的测试笔接于被测电路，在荧屏上即可显示出多少伏。改变被测电压，荧屏上的显示也跟着改变。当要对缓慢地一次性变化的电压进行扫描时，按下“RUN DV—AXI S”命令后，即见到荧屏上出现一直角坐标。以后每隔一定时间，在坐标系上打出一个点，以表示电压的高低。经一段时间后，这些点便连成“电压——时间”规律曲线。注意，所取时间间隔的长短，可在程序中的空循环语句中修改而定。

本机量程是直流电压0～2.6伏。精确度是将量程等分为约150个不连续的读数。如要扩大量程，可用多个分压电阻串联来代替R1；如要缩小量程，则要改变R4和R5的比例关系。公式是

$$U_{量程}=\frac{R_5}{R_4+R_5}\times2.6伏$$

及$R_4+R_5\approx1.2K$

这时要重新调节W，调法同上。

三、其它用法与存在问题

本机所利用的接口，本来是接一个可变电阻。所以，可直接利用它来测电阻，当输入近似于数字电压表的程序时，就成为一个数字电阻表。量程为1至130K左右。

如果在本机的输入端加接将温度转变为电压的电路，还可以作为数字测温表。

本文所用的方法，虽制作步骤多，但使用灵活多变。缺点是搬动不方便。如改为插件板，并用汇编语言，可进一步改善。

在设计调试过程中，得到暨南大学许世延教授的指点帮助，在此表示感谢!

附：(一)数字电压表程序（文件名定为DV－METER）

(二)“电压——时间”坐标曲线程序（文件名定为DV- AXIS）

DV－ METER

```
5 F = 0: HCOLOR= 3: ROT= 0
8 B =  PDL (0):C = B * 0.0147059
     1.0294118
12  HGR2 : IF B = 255 THEN 85: IF
     B < 53 THEN 90
15 C1 =  INT (C):D = 30: ON C1 +
     1 GOSUB 35,40,45,50,55,60,65
     ,70,75,80:: HPLOT 75,130 TO
     75,135
20 C2 =  INT ((C - C1) * 10):D =
     80: ON C2 + 1 GOSUB 35,40,45
     ,50,55,60,65,70,75,80
25 C3 =  INT ((C - C1 - C2 / 10) *
     100):D = 130: ON C3 + 1 GOSUB
     35,40,45,50,55,60,65,70,75,8
     0
30 C4 =  INT ((C - C1 - C2 / 10 -
     C3 / 100) * 1000):D = 180: ON
     C4 + 1 GOSUB 35,40,45,50,55,
     60,65,70,75,80:: HPLOT 180,5
     6 TO 204,135 TO 228,56: GOTO
     92
35  HPLOT D,56 TO D + 39,56 TO D +
     39,135 TO D,135 TO D,56:: RETURN

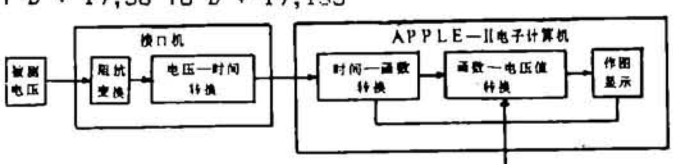
40  HPLOT D + 19,56 TO D + 19,135
     :: RETURN
45  HPLOT D,56 TO D + 39,56 TO D +
     39,90 TO D,100 TO D,135 TO D
     + 39,135:: RETURN
50  HPLOT D,56 TO D + 39,56 TO D +
     39,90 TO D + 30,95 TO D + 39
     ,100 TO D + 39,135 TO D,135:
     : RETURN
55  HPLOT D + 39,105 TO D,105 TO
     D + 30,56 TO D + 30,135: RETURN

60  HPLOT D + 39,56 TO D,56 TO D,
     90 TO D + 39,90 TO D + 39,13
     5 TO D,135: RETURN
65  HPLOT D + 39,56 TO D,56 TO D,
     135 TO D + 39,135 TO D + 39,
     90 TO D,90: RETURN
70  HPLOT D 56 TO D + 39,56 TO D +
     10,135: RETURN
75  HPLOT D,56 TO D + 39,56 TO D +
     39,90 TO D,100 TO D,135 TO D
     + 39,135 TO D + 39,100 TO D
     ,90 TO D,56: RETURN
80  HPLOT D + 39,95 TO D,95 TO D,
     56 TO D + 39,56 TO D + 39,13
     5 TO D,135: RETURN
85  HPLOT 0,75 TO 50,95 TO 0,115:
      HPLOT 60,56 TO 60,125 TO 70
     ,135 TO 100,135 TO 110,125 TO
     110,56: HPLOT 115,135 TO 115
     ,125 TO 120,130 TO 125,125 TO
     125,135: GOTO 92
90  HPLOT 50,75 TO 0,95 TO 50,115
     : HPLOT 70,56 TO 60,66 TO 60
     ,125 TO 70,135 TO 100,135 TO
     110,125 TO 110,66 TO 100,56 TO
     70,56
92  FOR E = 1 TO 100: NEXT E: GOTO
     8
```

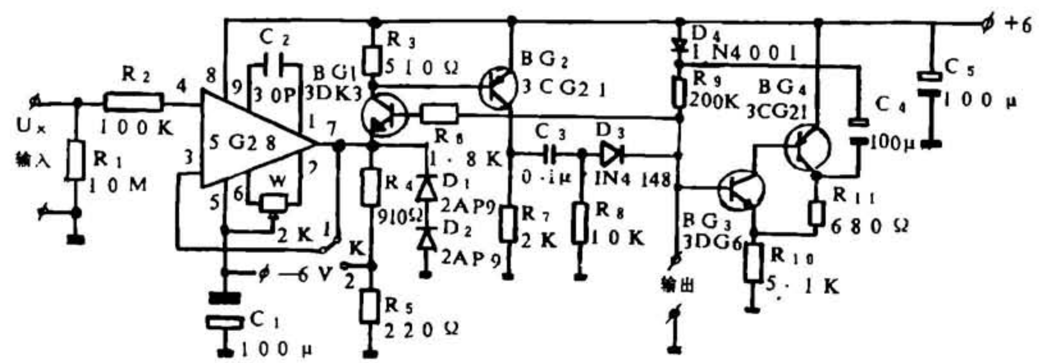
DV- AXIS

```
10  HGR2 :HCOLR = 3: ROT= 0: HPLOT
     3,9 TO 0,9 TO 0,191 TO 279,1
     91
20  HPLOT 5,0 TO 9,0 TO 9,4 TO 5,
     4 TO 5,8 TO 9,8: HPLOT 12,6 TO
     12,8: HPLOT 19,0 TO 15,0 TO
     15,4 TO 19,4 TO 19,8 TO 15,8
     : HPLOT 22,0 TO 24,8 TO 26,0

30  FOR A = 1 TO 279
40 B =  PDL (0):C = 191 - (B - 13
     ) * 182 / 242
50  HPLOT A,C
60  FOR E = 1 TO 100: NEXT E
70  NEXT A: END
```

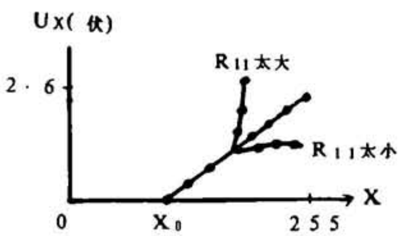


图一

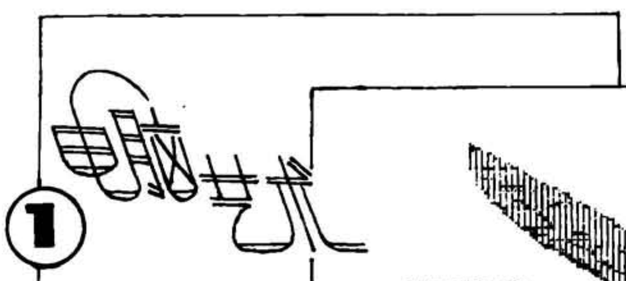


图(二)　　图中K接1 时，量程为0 至2.6 伏
K接2 时，量程为0 至0.5 伏

被测电压(V)	函数值(X)
U_1	X_1
U_2	X_2
U_3	X_3
U_4	X_4
U_5	X_5



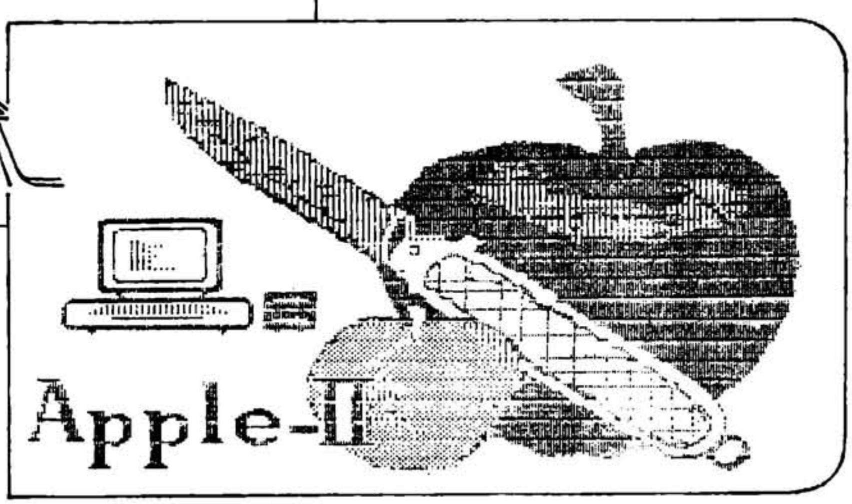
图(三)



APPLE机绘制

滋味

武汉市　秦华山



条件式的功能与应用

四川省南充市一中微机组　陈庆祥

一.问题的提出

例1. 从给定的n个数中统计出：其中正数的个数及它们的和、负数的个数及它们的和、零的个数。

这是一个常见而简单的例题。凡是学过基本BASIC的读者都知道：应该使用条件转移语句来编写程序，现在我们要求不使用任何控制程序流向转移的语句（包括开关语句）来解决它。这大概很难办到吧！请看下面的论述。

二.关系表达式与逻辑表达式

1. 关系表达式。

众所周知：条件转移语句是BASIC语言中使计算机具有逻辑判断能力的重要基本语句。

其一般基本形式为：

<行号>　关系表达式　THEN<行号>

在该语句中，关系表达式是实现语句功能的基本条件，是语句中的关键部位。

关系表达式的一般形式为：

<表达式1>　关系运算符　<表达式2>

其中表达式1、2应属同一类型的表达式，即同为数字型或是同为字符型的。关系运算符可为=、>、<、>=、<=、<>这六种之一。其运算过程为：先计算出两个表达式的值，然后用关系符把它们进行比较，判断这三者建立起来的关系是否成立，其结果为逻辑值：真（TRUE），用数字1表示；假（FALSE），用数字0表示。这说明关系表达式本身就代表一种判断。条件语句再根据以上结果进一步作出是否实现转移的判断，最后予以实施。

弄清以上道理后，我们完全可以在许多场合下撇开条件转移语句，直接使用关系表达式进行判断，同时还可以把其结果作为一般的数来参与算术表达式中的运算，从而使程序简洁，运算时间较短。至此，例1提出的问题和实现的要求便很容易解决。（请看程序1）

2. 在BASIC语言中还有另一种表达式——逻辑表达式，这也是在条件语句中常见的表达式。

其一般形式为：

<表达式1>　逻辑运算符　<表达式2>

这里的两个表达式仍是同一类型的。逻辑运算符有NOT、AND、OR三种，分别用以对两表达式作逻辑非、逻辑与、逻辑或等三种逻辑运算，其结果同样是逻辑值：真（用数字1表示）、假（用数字0表示）两者之一。不难理解，这种逻辑运算也是一种判断过程，因此我们同样可以用它来直接处理一些判断问题并把它融于算术运算之中去。

例2. 从三个互不相等的数中选出最大的一个数。（见程序2）

三.进一步的讨论

上述的关系表达式和逻辑表达式都可作为条件转移语句中实现转移的条件，故我们常把它们合称为条件式。实际上，有时条件式是由关系表达式与逻辑表达式混合而成的。由于条件式的运算结果是数字1或0，因此我们完全可以把它纳入算术运算，使之成为混合型的表达式（当然，除条件式这一部分之外的内容，应都是数字型的表达式）。这种混合型的表达式兼有逻辑判断和算术运算的功能，可以出现在条件转移语句、赋值语句、打印输出语句等语句中。合理地使用条件式，可以使程序简化。这种平时不引起人们注意的条件式，在处理一些头绪繁多的复杂的逻辑推理过程中更能大显身手，成为特别有用的工具。这时用它来编写程序，可以使推理过程通达、直观、还不容易出错。（这是因为它能大大减少程序流向的转移的缘故）。这种按照问题本身提出的条件顺序进行逻辑推理犹如顺藤摸瓜，符合人们的一般思维方式。

不过，需要注意的是：在书写混合型的条件式时，应特别小心运算顺序。为了避免混淆，可以按

推理顺序适当添上些括号，即容易识别。

下面是一个十分有趣而难度较大的例题。读者可以试一试用条件转移语句来设计的难度。

例3. 公安人员断案

公安人员审问甲、乙、丙、丁四个嫌疑犯，已确知，这四个人当中仅有一人是偷窃者，还知道这四个人的说话，不是完全诚实者，就是完全说谎者。在回答公安人员的问话中：

甲说："乙没有偷，是丁偷的。"

乙说："我没有偷，是丙偷的。"

丙说："甲没有偷，是乙偷的。"

丁说："我没有偷，我用的那东西是我家里的。"　据上述四人答话，请判断谁是偷窃者。

设计思路：

用A、B、C、D表示甲乙丙丁四个人，变量的值是1或0。1表示偷了，0表示没有偷。用A1、B1、C1、D1表示甲乙丙丁四个人答话，其值是1表示诚实，是0表示就说谎。我们用以A、B、C、D四个变量为循环变量的四重循环排出四人偷窃与否的所有情况，（各层循环变量仅取值0、1、）再用四个条件式作为A1、B1、C1、D1的取值。注意条件为：完全诚实或完全说谎时为真，即各人答话全是真话或全是谎话时，该变量值为1，除此之外都为假，值是0。当A1、B1、C1、D1的值全为1时，当时的A、B、C、D的值中，为1者便是偷窃者。（不过，丁的答话对判断过程无影响，其值可给为1。）

(以下三程序均在APPLE－II型机上通过)

程序1.

```
5  REM  PROGRAM 1.
10  INPUT "N=";N
15  FOR I = 1 TO N
20  READ X
30 S1 = S1 + X * (X > 0):N1 = N1 +
     (X > 0)
40 S2 = S2 + X * (X < 0):N2 = N2 +
     (X < 0)
50  NEXT I
60  PRINT "(+):";N1,"T(+):";S1
70  PRINT "(-):";N2,"T(-):";S2
80  PRINT "(0):";10 - N1 - N2
90  END
100  DATA 3,-4,0,24.5,-11,-5,7,0,
     11,6
```

```
]RUN
N=10
(+):5            T(+):51.5
(-):3            T(-):-20
(0):2
```

程序2.

```
]
]LIST

5  REM  PROGRAM 2.
10  INPUT "A,B,C=";A,B,C
20  PRINT : PRINT "MAX:";
30  PRINT A * (A > B AND A > C) +
     B * (B > A AND B > C) + C *
     (C > A AND C > B)
40  END

]RUN
A,B,C=25,-44,17

MAX:25
```

程序3.

```
5 REM GONG AN REN YUAN DUAN AN.
10 FOR A=0 TO 1: FOR B=0 TO 1
20 FOR C=0 TO 1: FOR D=0 TO 1
30 LET A1=((B=1 AND D=0) OR (B=0 AND
D=1))
40 LET B1=((B=1 AND C=0) OR (B=0 AND
C=1))
50 LET C1=((A=1 AND B=0) OR (A=0 AND
B=1))
60 LET D1=(D=1 OR D=0)
70 IF A1+B1+C1+D1=4 THEN 100
80 NEXT D,C,B,A
90 GOTO 110
100 PRINT"A B C D": PRINT A; " "; B; " "; C; " ";
D
110 END

 RUN

A B C D
 0 1 0 0
```

（乙为偷窃者）

APPLE Ⅱ绘图技巧

四川省重庆一中高一(1)班　谢志坚

APPLE Ⅱ微型计算机目前在国内十分流行，尤其深受广大中学生的喜爱。在APPLE Ⅱ浮点BASIC语言状态下，能够使用两页（PAGE）高分辨率绘图区，给使用带来了方便。但由于占用较多内存，因此绘图程序不能太长，以免画面被破坏。另外，浮点BASIC语言绘图速度太慢，在程序中使用一些技巧，既提高绘图速度，又使程序简短，画面更生动、美观，就显得十分必要。这里向大家介绍几种简单的绘图技巧。

一．反相处理

高分辨率图形打印中有一种叫反相打印，即图形上的一个亮点在纸上输出一个空白点，而图形上的一个暗点在纸上输出一个黑点。这样的打印使图形在纸上呈底片状，特别适合打印比较单调的图形，如零件图等。

在屏幕上使用反相处理显示将得到反相打印的效果，使线条更清晰，画面更生动。

程序结构如下：

```
10  HGR：HOME （或HGR2：HOME）
20  HCOLOR=7
30  HPLOT  0，0：CALL  62454
40  HCOLOR=0
50  HPLOT   ……
60  HPLOT
 .
 .
 .
```

程序中10语句设置第一页或第二页状态，20语句设置屏幕背景颜色（当然可以是其它颜色），30语句用设置好的颜色涂背景，40语句设置所绘图形的颜色黑色，50语句以后开始绘图。

30语句中的0，0可以换成合理范围内的任意数，如279，159等；40语句如果不设置黑色，就没有反相效果，但仍可以使画面更生动，不单调、呆板。

下图是经过反相处理后的图形：



二．移位处理

有时候同一图形需在不同的位置上绘出，这就需要作移位处理。

移位处理往往是以某一位置的图形（或图形上一点）作参照，其它位置上的图形就要用相对位移后的坐标来画。

如下图的这只仙鹤，以它嘴尖一点为参照，在后上方再画一次，成为两只飞行的鹤，就可采用这样的程序结构：

```
10  HGR：HCOLOR=7
20  X=0：Y=0：GOSUB  50
30  X=30：Y=-25：GOSUB  50
40  END
50  HPLOT   57+X，100+Y   TO   69+X，
     106+Y   TO  ……
60  HPLOT  ……
 .
 .
990  RETURN
```

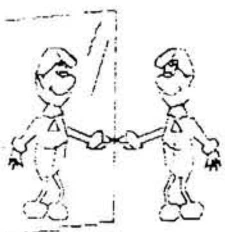
程序中20、30语句实际是设置了位移参数，调用50、60…990绘图子程序。当X=0，Y=0时，就在原位置画出图形，当X=30，Y=-25时，就后移30，上移25画出图形。



利用这种方法，只需设置不同的位移参数就可画出许多同样的图形来。读者想画一幅百蝶图吗？利用随机函数产生位移参数，你一定会觉得乐趣无穷的！

三.对称图形处理

生活中的对称图形不少，想要描绘对称图形，如果按部就班地画，显然会浪费一半语句，因为根据对称图形的特点，只需画出一半图形，另一半图形让计算机根据公式计算各点坐标画出。

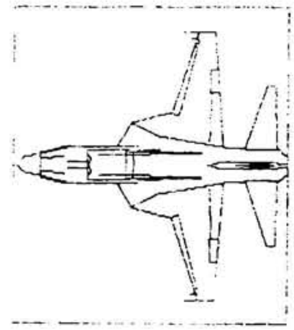


看上面这幅“蓝精灵照镜子”多有趣！我们只要画出左、右任一个蓝精灵，再添上镜子就行了。（这里假设镜右框为中心线139，0－139，159）

程序结构如下：

```
10 HGR：HCOLOR＝6
20 ONERR  GOTO  40
30 READ  X1，Y1，X2，Y2：HPLOT  X1，
   Y1 TO X2，Y2：HPLOT 139－（X1－139）
   ，Y1 TO 139－（X2－139），Y2：GOTO 30
40 END
50 DATA  150，57，169，69，……
60 DATA  ……
 .
 .
 .
```

此程序用DATA语句提供图形各端点坐标，30语句读出这些数，先在原位绘图，然后在关于中心轴对称的位置上画图。20是用在数据读完后，自动结束程序，避免出现“OUT OF DATA ERROR”的错误信息。改动30语句中的公式，让图形上下对称，不也很有趣吗？见下图：



四.对称移位和中心对称处理

参见二、三。对称移位就是在对称后再移动位置，这实际上也是一种中心对称。程序结构和三相同，只需改变计算公式即可。

五.交互技术

利用POKE指令可以将两页图形显示出来。如果两页图形有某些关联，利用视觉保留现象，将使你得到奇特的效果。

程序结构如下：

```
10 HGR：HOME：HCOLOR＝7：HPLOT ……
 .
 .
100  HGR2：HOME：HCOLOR＝7：HPLOT
……
 .
1000  POKE  －16304，0：
      POKE  －16301，0：
      POKE  －16300，0：
      POKE  －16297，0：
      POKE  －16302，0：
      POKE  －16299，0：
      GOTO  1000
```

先在两页上画好图形，然后连续不断地交互显示两页图形。交互技术可分两种：

1.交互重叠式

实际是将两页图形合在了一起，（利用视觉保

COMX PC1 BASIC中CHR$ 函数的妙用

——安徽省马鞍山市第一高级职业中学85财会班 马剑——

由于COMX PC1微电脑的音响功能较好，色彩丰富，价格又低于APPLE Ⅱ等高档机，所以被很多中小学校用于电脑教学。

COMX PC1的BASIC有很多特殊功能，我这里只介绍CHR $函数在游戏中的妙用。

一.在动画游戏中快速移动一个大图形

COMX PC1 BASIC的一个大缺点就是打印字符速度慢。最难办的是移动一个由几行字符拼成的图形，它们一移动，常常被分成了几个部分先后移动，即上下两行字符不能同时移动。例如在有的游戏程序中画出的大炮是由两行字符，即炮筒和炮身拼成的，移动它，只能先将炮筒向左或右移一格，然后才能将留在原地的炮身移过去，十分明显的把大炮割成了两半。

如何解决这个问题呢？

在COMX PC1的ASCII码中，有四个特殊的代码：128，129，130，131。

用PRINT CHR $（ ）打印它们，它们就有用处了。128是将光标向上移一行；129是向右移一格；130是向下移一行；131是向左移一格。例如：

PRINT CHR $（65，128，66，130，67）

这个命令被执行后，先打印出字母A，然后在A的右上边打印出字母B，最后又在字母A的那行上打印出字母C。

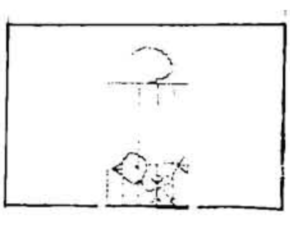
这是因为打印字母A后，代码128将光标向上移了一行，所以字母B打印在字母A的右上方。随后，代码130将光标移回了原来的所在行，打印出字母C。

怎么利用这四个特殊的代码来移动图形呢？

请看程序1，这个程序运行后，屏幕下方出现两个字母：A和H。字母A在字母H的上面。按方向键中的左右键（COMX 35上是操纵杆），字母A和字母H同时向左或右移动，虽然不在同一行上，但移动后，看不出被分成两次移动的情况。这个功劳就是第三个语句的PRINT CHR $的。

CHR $函数的第一个数据是131，将光标向左移一格，打印一个空格，把图形右移时留下的字符擦掉，这时光标已经回到原处，打印出字母A，

留现象）但两页图形并没改变。如下图，在第一页上画一个鸟笼，第二页上（鸟笼内位置）画一只小鸟，显示出的效果就是小鸟在鸟笼里，但画面有较大的闪烁感 。



2.交互动画式

根据动画原理，两页图形大致相同，只有微小变化，那么通过显示，就会得动画的效果。

例如，一页上画张开的嘴，一页上画闭着的嘴，显示效果就是嘴一闭一合，象说话似的。

看到这里，读者或许会问：如果我只有保存在磁盘中的源图形库（BSAVE进磁盘的），又该怎样处理呢？这一点请参见《电脑》一九八七年第三期《高解析度图形剪辑程序》。

再用一个空格把图形左移时留下的字符擦掉。然后，三个131又将光标移到A左边的空格上，130使光标向下移一行打印出空格、字母H、空格。两个空格也是为了擦掉图形左或右移时留下的字符。

在COMX PC1 BASIC中，对PRINT CHR $的解释速度快于PRINT″ ″，因此执行PRINT CHR $比执行PRINT″ ″的时间短得多，所以字母A和字母H虽然是先后移动的，但先后隔的时间短得让人看不出来。

用这种方法移动图形，只要每行的字符不多，它甚至能移动占三四行字符的图形。

二.用CHR$ 为字符扩充四种颜色

用CHR $和SHAPE命令配合，可以使字符颜色达到八种。

在用SHAPE命令定义字符图形时，八个二进制数字的前面两个数字用于指定颜色：00为黑；01为蓝；10为绿；11为青。

在用PRINT″ ″打印字符时，颜色只有这四种，很单调，但如果将你定义字符的代码加上128（代码必须大于15），用CHR $函数输出时，那两个二进制数字代表的颜色就变了：00为红；01为洋红；10为黄；11为白。

如果你的游戏程序需要多种颜色，就可以用CHR $函数调出四种鲜艳的暖色和四种冷色配合，把你的游戏程序变得更加迷人。

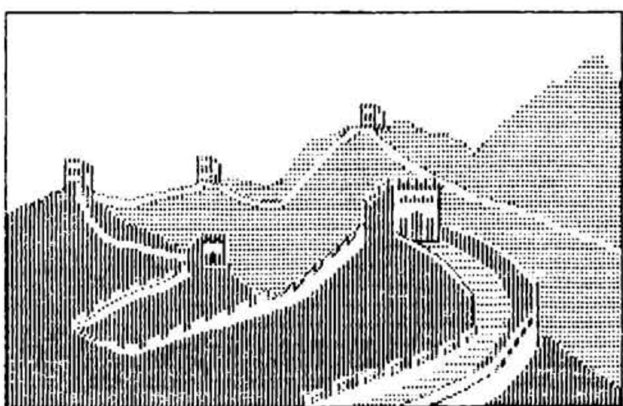
见程序2，它将打印出八种颜色的彩色方块。其中缺了的方块是黑色的。

程序1.

```
10 CPOS(10,0):CLS
20 E=18:F=20
30 CPOS(E,F):PRINT CHR$(131,32,65,32,
   131,131,131,130,32,72,32)
40 K=KEY:IF K=137 IF F<36 THEN F=F+1
50 IF K=139 IF F>1 THEN F=F-1
60 GOTO 30
```

程序2.

```
10 CPOS(0,0):CLS
20 SHAPE(65,"3F3F3F3F3F3F3F3F3F")
30 SHAPE(66,"7F7F7F7F7F7F7F7F7F")
40 SHAPE(67,"BFBFBFBFBFBFBFBFBF")
50 SHAPE(68,"FFFFFFFFFFFFFFFFFF")
60 PRINT CHR$(65,66,67,68)
70 PRINT CHR$(67,68,195,196)
80 PRINT CHR$(65,66,193,194)
90 PRINT CHR$(193,194,195,196)
100 WAIT(100)
```



汕头大学化学系　李　丹



打印上下标的一种方法

汕头市教育局电教站　　蔡希人

编写辅助教学软件时常须用数字做各种符号的上下标。利用超级中文卡主盘运行本程序，即可达到目的。

例如：要打印$Ca(OH)_2+CO_2=H_2O+CaCO_3\downarrow$

可在键盘打入：

? "Ca (OH) CTRL-F-1 2 CTRL-F-0 +CO CTRL-F-1 2 CTRL-F-0 =H CTRL-F-1 2 CTRL-F-0 O+CaCO CTRL-F-1 3^ CTRL-F-0"

就是说，要打印下标前，先打CTRL-F-1，后打相应数字即可。再打CTRL-F-0即恢复原字符输出。同理，要打印上标时，只要把CTRL-F-1键换成CTRL-F-2键即可。

本程序可做为总程序的开始部分，也可以用它代替超级汉卡程序主盘的欢迎程序，使开机即自动备好上下标符号。还可用下面的办法直接写入磁盘：

1. 用去除禁写封口的超级汉卡主盘起动机器。

2. 运行本程序。

3. CALL—151进入监控状态。

4. 打入8D00<8A00.8CFFM回车。

5. CTRL—C进入BASIC。

6. 打入BSAVE SETF，A$8D00，L$300回车。

7. 用8700.89FF代替8A00.8CFF重复4，5。用SETG代替SETF重复6。

也可直接打入GOTO 7。

机器原来使用的是字符集SETA. 如果希望机器经常出现某一字符集α 可打入BLOAD SETα (回车)，此后机器即采用新的字符集。要恢复原状，只需再打BLOAD SETA (回车)

每个字符集约占内存三页。起动时，机器中已存入A、F、G、J、K、L、R等七集。占用了7B00H到8FFFH约5.5K的内存。HIMEN定为7CFFH。如果您怕内存不够用，可另定HIMEN，当然此时能用的字符集也减少了。

每个字符占八位内存。字符形状的定义方法是：把每个字符分解成七列八行，发光的点置为1，不发光的点置为0，再在每列的最后加上0或1，(可控制色彩)，然后把每列依从右到左的顺序做为一个8位二进制数。依从上到下的须序逐次存入该字符相应的内存位置。

例如：A集中的字符O的ASCII码是4FH，其定义从8D00H+(4FH−20H)*8=8D00H+178H=8E78H处开始存入，内容如下：

存贮位置	二进制数	十六进数
8E78H	01111100	7C
8E79H	10000010	82
8E7AH	10000010	82
8E7BH	10000010	82
8E7CH	10000010	82
8E7DH	10000010	82
8E7EH	01111100	72
8E7FH	00000000	00

掌握了这个规律，您不难任意创造自己需要的字符。

```
1 I = 34648:I1 = 2:I2 = 4: GOSUB 4
:I = I + 8: GOSUB 4:I = I + 24:I1
= 4:I2 = 2: FOR K = 1 TO 9: GOSUB
4: NEXT :I1 = 2:I2 = 4: FOR K = 1
TO 2:I = I + 8: GOSUB 4: NEXT :I =
 I + 256: POKE I,4: POKE I + 1,14:
 POKE I + 2,2  FOR K = I + 3 TO I
 + 7: POKE K,4: NEXT : RESTORE
2 I = 35416:I1 = 2:I2 = 4: GOSUB 5
:I = I + 8: GOSUB 5:I = I + 24:I1
= 4:I2 = 2: FOR K = 1 TO 9: GOSUB
5: NEXT :I1 = 2:I2 = 4: FOR K = 1
TO 2:I = I + 8: GOSUB 5: NEXT :I =
 I + 256: FOR K = I TO I + 4: POKE
 K,4: NEXT : POKE K,21: POKE K + 1
,14: POKE K + 2,4
3 HOME : VTAB 7: PRINT "CTRL-F 0:
": PRINT "     1 2 3 4 5 6 7 8 9 +
; = - ^": PRINT : PRINT "CTRL-F 1:
": PRINT "     1 2 3 4 5 6 7 8 9 .
.. -- - ↓": PRINT : PRINT "CTRL-F 2:
```

(下转第47页)

关于dBASE Ⅲ子串函数的巧妙用法

广州军区后勤部自动化站　邓晓琼

在应用dBASEⅢ编程时，常遇到字符之间的转换（如：1→一、2→二、……）和由用户定义的某些代号表示的中文字符串还原（例：T→吞吐量、G→供应量）问题，为了给用户提供方便，常在选中某个代号后要求立即显示或打印出它所代表的字符串，此时就有字符串的选择方法问题。有些人采用条件语句逐个判断分别赋值的方法，这使程序长度增加，且多重判断造成的程序冗余度大。笔者经实践总结出一个算式，是选取子串的较好方法，即灵活地运用子串函数并加以适当的算法组成表达式：

substr (w, ((asc (char) −n) × m+p+1) , b)

其中：substr (……) 为子串选择函数；

w：表示用户定义的字符串，它是用户所需全部（若干组同长度）子串的集合，注意每个汉字是占两个ASCII码字符位；

b：表示用户所需选取字符串的长度；

asc ( )：求括号内字符的ASCII码值函数；

char：表示一字符变量，它的变化范围应是有序的，且其每一个确定的值对于w字符串中的某个子串，一般为该子串的代号（由用户定义的）或某种转换字型；

n：是w字符串中排在第一的子串所对应char字符的十进制ASCII码值；

m：是一数字，它表示子串的长度。因各子串字符个数相同，则长度为定值；

p：表示附加位数字，是用户在某些情况下，不需要子串的前（或后）一部分的字符长度。

也就是说，首先由 (asc (char) −n) 计算出所选子串是排在w字符串中从0开始的第几个子串；再通过 ((asc (char) −n) ×m+1) 算出所选子串的开始位置，但当用户只需使用所定义子串的后（或前）一部分而不需要整个子串时，那部分不需要的字符长度就是附加位p的值，则通过确定算式 ((asc (char) −n) ×m+p+1) 的值计算出来的才是选中字符串的真正开始位置。

值得注意的是：①一旦w字符串确定，则在此算式中，n、m、p、b都为定值，而char为变量。②此式不适用于文本文件。因为此式要求用户定义的若干字符代号与其对应的ASCII码值都是连续的，而在IBM PC系列机上，能做到这点的最多是26个英文字母A～Z或a～z，它们对应ASCII码值的十进制表示分别是65～90、97～122，所以除了每个w字符串的字符数最多可有254个的条件外，其中所包含的子串个数也受到限制，最多只能是26个。一般常采用英文大写字母和字符数字0～9作代号。

下面举例说明这一算式的具体用法，为了演示方便，仅编写了一些短小的程序段来说明问题，所列程序均在IBM PC／XT微机上通过。

笔者在研制《物资微机管理系统》软件时，为处理打印单据、财务发票等，编写了一段将阿拉伯数字表示的金额转换成中文大写的子程序，认为它短小精悍、方法简单，借此推荐给广大读者交流。

程序原理：首先设置两个字符串（w1、w2）分别存放中文大写数字和金额单位，把阿拉伯数字表示的金额数乘以100化为整数后送入一字符串（JE）中，然后按从低位到高位的顺序逐一取出数字并转换成相应的中文大写和金额单位，转换方法采用了前述的表达式，再将转换结果依次加入另一字符串（HJE）中，直到最高位为0时止，此时，便可把表示该金额中文大写的字符串用于输出了。这里，完全按发票要求，即去掉最高位为0的项、对低位为0的项要填入“零”表示。

根据使用单位的要求，程序只考虑了金额不超过亿的情况。为表演方便，将原程序中的金额数目改由键盘输入。

程序一

```
. TYPE T1.PRG
* 显示出所选代号表示的汉字名程序段
W='A  高级工程师 B-  工 程 师  C  助理工程师
@ 2,1 SAY '职称代号：' TW
```

```
WAIT '             请选择代号(A,B,C)    ' TO CHAR
CHAR=UPPE(CHAR)
RRR=SUBS(W,((ASC(CHAR)-65)*14+4),10)
* 上式中,a=ASC(A)=65,m=14,p=3,b=10
@ ROW()+1,11 SAY '职    称    是：'+RRR
RETU

  DO T1
 职称代号：A--高级工程师 B-- 工 程 师  C--助理工程师
           请选择代号(A,B,C)：B
           职    称    是：工 程 师
```

程序二

```
. TYPE T2.PRG
* 将阿拉伯数字表示的年度转换成中文数目字
W='〇一二三四五六七八九'
ND=SPACE(4)
@ 2,2 SAY '请输入年度：' GET ND PICT '9999'
READ
IF ND=SPACE(4)
  RETU
ENDI
L=1
N=' '
DO WHIL L<5
  N=N+SUBS(W,((ASC(SUBS(ND,L,1))-48)*2+1),2)
  * 上式中,a=ASC(0)=48,m=2,p=1,b=2
  L=L+1
ENDD
@ 3,15 SAY N
RETU

  DO T2
  请输入年度：1987
              一九八七
```

程序三

```
. TYPE T3.PRG
* 将阿拉伯数字表示的金额转换成中文大写
W1='零壹贰叁肆伍陆柒捌玖'
W2='仟佰拾万仟佰拾元角分'
JE=0.00
HJE=' '
@ 2,2 SAY '输入金额(小写)：' GET JE PICT '99999999.99'
READ
L=11
JE=STR(JE*100,10)
DO WHILE L>0
  L1=SUBSTR(JE,L,1)
  IF L1=' '
   EXIT
  ENDI
  T=SUBSTR(W1,(ASC(L1)-48)*2+1,2)
  L=L-1
  T=T+SUBSTR(W2,(ASC(STR(L,1))-48)*2+1,2)
  HJE=T+HJE
ENDD
IF SUBS(HJE,1,2)='零'
  N=LEN(HJE)
 'HJE=SUBS(HJE,5,N-4)
ENDI
@ 3,2 SAY '金    额   (大写)：'+HJE
RETU

  DO T3
  输入金额(小写)：   823451.97
  金    额   (大写)：捌拾贰万叁仟肆佰伍拾壹元玖角柒分
```

中国软件行业协会DG应用协会首届年会在苏州举行

中国软件行业协会DG应用协会首届年会已于一九八七年四月二十二日至二十四日在苏州举行。

本次年会共收到有关美国DG公司的MV超级小型机和Desktop微型机的软件产品成果和论文四十余篇。会上除成果和论文交流外，还进行了会员单位之间的对口交流与协作，由TDJ电脑集团做了咨询，帮助会员单位解决问题。参加会议的有一百五十多人，其中包括国内著名的专家和教授，会议还邀请了亚洲DG公司、香港创建基立有限公司和国际庄明企业公司代表介绍它们的新产品，并由亚洲DG公司高级系统工程师丘国梁先生介绍了AOS／VS操作系统新版本和硬软件情况。

会上还介绍了我国自行研制的8M内存扩充板和终端等硬件产品，可以提供MV系列和用户之用。Desktop计算机已分析完毕，正准备着手国产化工作。

在年会上对优秀成果和论文颁发了奖状，表扬了DG机应用方面的优秀成果和科技人员，鼓励开发出更多更好的软件产品，满足国内各单位应用。

在年会期间召开了理事会，决定在北京成立北方技术咨询部，在上海成立南方技术咨询部，以便密切会员单位之间关系，加强交流与协作，克服重复开发，提高软件开发与应用水平，并开展技术咨询服务工作。（钱承德）

电脑的选购、验机和投入使用要点

商业部安徽商业管理干部学院

教研部　陈庆章

许多准备开展电脑应用的单位首先感到棘手的工作是：面对种类繁多的电脑，究竟选购哪一种为好；已买来的机器是否可正常运行，无故障；在投入使用时应注意些什么。由于对这些问题并不十分清楚，结果导致不少麻烦，甚至造成严重损失。如我省某市统计部门，投资八万元买进一套电脑系统，但由于投入使用时不慎，加电时竟使主机烧毁；还有许多单位未加认真考虑，盲目购进机器，在以后使用中又感觉性能、价格等并不合算，或维修、服务困难等。笔者曾经为不少单位选购机器，验机和安装使用，或作咨询。本文就此谈一点体会。

一　选购

开展电脑应用，首要任务就是选定机种。目前，市场上各种型号电脑品种众多，要选择得好，也决不是一件易事。作为使用单位首先要明确：

·引进电脑目的如何

·使用后它的经济效益

·处理的业务范围和信息流的情况

·资金和电脑应用人才情况

这一步的目的在于对电脑要处理的信息及其量有初步的概念，对要求电脑做的事情胸中有数，对领导的决策、单位资金和人才清楚。

接下来就是对机种的选定。选择机种总的原则是：性能／价格比较好，处理能力能满足要求，互换性好，容易扩充，培训、维修有保证，软件丰富。对于一般并不十分精通电脑的购机者，通常可从以下几点考虑：

1. 优先选择国家推选的重点机型。这些机型都是国际国内流行广泛，经过实践证明的优秀机种，有丰富的软件支持，有强大的服务网。例如长城0520、紫金一Ⅱ以及和IBM—PC兼容的其它机种。

2. 电脑的功能强大与否，在很大程度上取决于软件。所以，电脑软件丰富与否，是否有自已适用的应用软件是购机时应着重考虑的。它直接关系到应用工作的开展，选择好了将会收到立竿见影的效果。购买软件，一般要考虑软件的适应性、兼容性、升级换代能力，排错能力，软件研制厂家等。

3. 电脑厂商是否拥有较强的技术力量和分布广泛的服务、维修机构。特别是购机单位本地区或邻区有没有这样的机构。其培训业务开展如何。

4. 硬件和软件的图纸资料是否齐全。备件，扩充件供应是否充足，有关资料出版情况如何。

5. 电脑的汉字处理功能。

6. 根据应用目的不同，选择机型也应不同。如若应用于企业管理和教学，则希望配备多功能多用户系统，要求机器有丰富的外围接口。对于简单的过程控制和仪表控制，则选用单片微处理器和单板机就可，当然还要考虑I／O通道。

7. 在硬件系统指标满足要求的情况下，应选择用户比较熟悉的机种。因为用户已积累了一部份使用经验和可利用的软件。把自已过去的成果考虑进去会加快电脑应用的开发。

二　验机

选好机型，在提货时应进行验机。验机前应查看一遍机器所附的操作手册及其它资料。下面以使用较多的IBM——PC机为例，谈谈验机方法：

1. 对整机及其部件的外观进行全面的检查，看一看有没有损坏。如果允许，可对电源、硬盘驱动器、软盘驱动器，及系统板和各种扩展板等主要部件进行逐块抽检。方法是让它们在经过鉴定的同型机器上运行。如果一切正常，则认为各部份正常。

2. 在系统正确地连接好后，就可加电检验。

先接通显示器和打印机电源，再接通主机电源。于是，系统开始自检。自检时间一般是13～90秒不等，此时有三个响应：

·显示器上出现光标

·检查完全部内存后，喇叭发出一短声“嘟”

·系统开始启动。听到盘片的旋转声，调入操作系统，这时显示器上显示：

```
Current date in the 1-01-1981
Enter new date
```

此时按回车键或者从键盘输入年、月和日后按回车键，显示器显示：

```
Current time is 0: 02: 34· 67
Enter new time
```

此时按回车键或者从键盘输入时、分和秒后按回车键，显示器显示：

```
The IBM Personal Computer DOS
Version 2·00 (c) Copyright IBM Corp
1981、1982、1983
A>
```

这表明已进入DOS状态。如果上述响应不正常，则系统主机有故障。

在出现提示符A>后，若先按Ctrl—P，然后每按一次回车，则打印机的纸走一行。如再打入命令DIR，则显示器和打印机同时输出盘片上所有文件的信息。这一过程可证明主机与打印机通讯良好。

当打入BASICA命令时，则显示器上显示：

```
The IBM Personal Computer BASIC
Version A1·10 Copyright IBM Corp 1981,
1982 61066 Bytes free
Ok
```

系统进入BASIC状态，可运行DOS盘的Samples程序。

顺利通过上面三步，一般可认为系统基本完好。

3．进一步的验机可在软盘驱动器中插入诊断程序盘，按Ctrl—Alt—Del键，系统重新启动进入诊断状态，对系统各部份进行检查。此时显示器显示一份“菜单”：

```
The IBM Personal Computer
DIAGNOSTIC
Version 2·00 (c) Copyright IBM Corp
1981, 1982
SELET AN OPTION
0—RUN DIAGNOSTIC ROUTINES
1—FORMAT DISKETTE
2—COPY DISKETTE
3—PREPARE FIXED DISK FOR
RELOCATION
```

以后，用户则可按系统在显示器上的提示进行检查。诊断程序使用的具体步骤，可以参见《电脑》杂志87年第二期。

4．诊断程序验机结束后，让电脑连续工作十二小时左右进行考机。

上述步骤全无异常，则认为机器一切良好。

三　投入使用

本人分析不少用户，其电脑发生故障的原因主要是由于安装使用上的错误所致。现提出以下几点供注意：

1．电网电压不能波动太大，应在210伏～230伏之间。一般应采用稳压电源。对于多台机器，若有能力，最好一台稳压电源给部份机器供电，以防稳压电源故障，而影响所有机器。

2．供电插座的零线插孔与火线插孔不能接反，应为标准接法。即面对插座正面，左边零线右边火线。地线插孔接地电阻要小于4欧，与零线插孔之间的电压要小于6伏。

3．开机顺序要先外设后主机，关机则反之。特别是在主机运行期间，不能拔掉主机与外设的连接电缆，否则，这种电扰动会严重影响机器的性能。

4．注意所用打印机是否与主机配套。如果不配套，应对连线电缆进行改动。

5．用诊断程序验机时，要注意诊断程序是否与机器完全兼容。特别是对一些兼容机，在不完全兼容情况下，验机过程中可能会出现某些错误信息，未必是机器故障。

6．对IBM—PC／XT验机结束后，应锁住硬盘，以免运输中震坏盘片。步骤是重新装入诊断盘，开机，显示第一份“菜单”，按3键，则硬盘驱动器面板上的指示灯闪亮一次，表明已锁住盘片。当再次加电开机时，系统自动解锁。

除以上各点外，当然在使用电脑中还要注意机房工艺卫生，使机房保持一定的空气条件如温度、

程序优化一例

河北省医学科学院药研室　杨　纯

清华大学出版的"BASIC语言程序自动化工程常用算法"（张巨洪等编）一书的101页，介绍了"产生任意均数和方差的正态分布随机数"的程序。其中计算随机数经验曲线的程序如下：

```
8070 FOR I=0 TO 6 STEP .3
8075 C=0
8080 FOR K=1 TO N
8085 IF X(K)>I GOTO 8100
8090 IF X(K)<I-.3 GOTO 8100
8095 C=C+1
8100 NEXT K
8105 PRINT I:TAB(10):C:TAB(20+2+C/10):"+"
8110 NEXT I
```

该程序是将介于－0.3和6之间的均数为3，标准差为1.5的2000个随机数X（K），按组距0.3分组并计算其频数，然后绘出分布示意图。若将程序改变如下，则可使运算时间减少90％。

```
8070 FOR K=1 TO N
8075 IF X(K)<-.3 THEN 8090
8080 J=INT(X(K)/.3+1)
8085 C(J)=C(J)+1
8090 NEXT K
8095 FOR J=0 TO 20
8100 PRINT J*.3:TAB(10):C(J):TAB(20+C(J)/5):"+"
8110 NEXT J
```

在IBM PC／XT机上运行原程序需5分5秒，改变后的程序仅用34秒。原程序对X（K）进行分组时，是将大于I和小于I－0.3的值剔除、保留介于I和I－0.3之间的值。故对每一组都需对2000个随机数逐一检验。由于有21组，共需检验42000次。而改变后的程序是对任一X（K）直接确定其组别，用C（J）记录每组频数，仅需检验2000次即可，故运行时间大为减少。

优化后的 程序清单如下：

```
8005 LPRINT "INPUT NUM.OF DATA N=?"
8010 INPUT N
8015 PRINT
8020 DIM X(N)
8021 DIM C(N)
8025 S=0
8030 FOR K=1 TO N
8035 GOSUB 8200
8040 X(K)=Y
8045 S=S+X(K)
8050 NEXT K
8055 M=S/N
8060 LPRINT TAB(15):"MEN.=":M
8065 PRINT :TIME$="0:0:0"
8070 FOR K=1 TO N
8075 IF X(K)<-.3 THEN 8090
8080 J=INT(X(K)/.3+1)
8085 C(J)=C(J)+1
8090 NEXT K
8095 FOR J=0 TO 20
8100 LPRINT J*.3:TAB(10):C(J):
     TAB(20+C(J)/5):"+"
8110 NEXT J
8115 LPRINT TIME$:END
8200 S1=0
8205 FOR I=1 TO 12
8210 S1=S1+RND(1)
8215 NEXT I
8220 Y=3+1.5*(S1-6)
8225 RETURN
INPUT NUM.OF DATA N=?
              MEN.= 2.989878
 0         19   +
 .3        28      +
 .6        43           +
 .9000001
           65                 +
 1.2       66                 +
 1.5       87                       +
 1.8       112                             +
 2.1       116                              +
 2.4       161                                     +
 2.7       146                                  +
 3         152                                   +
 3.3       157                                    +
 3.6       143                                  +
 3.9       153                                   +
 4.200001
           112                             +
 4.5       112                             +
 4.8       83                       +
 5.100001
           61                +
 5.4       53              +
 5.700001
           38          +
 6         29      +
00:00:47
```

相对湿度，洁净度等。

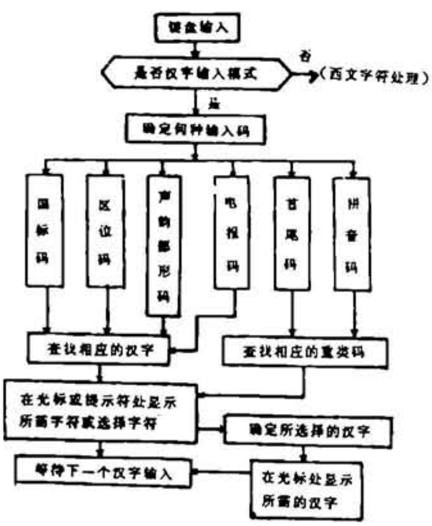
电脑是知识密集性产品，作为用户在选购，验机和投入使用时应慎重行事。最好能聘请有经验的专家指导，方可购到称心如意的产品，使用起来安全，正常。

运用程序语言处理汉字信息的一些方法

暨南大学电子技术班　陈汝祥

到目前为止，国内在IBM－PC计算机及兼容机上所使用的汉字系统都以早期“电子工业部第六研究所”的CC－DOS为基础或作借鉴而发展起来的。汉字库是以“中华人民共和国国家标准信息交换汉字编码GB2312－80”的汉字顺序排放。与此同时，结合IBM－PC计算机的特点，CC－DOS使用的汉字机器内码是一种高位码为“1”的两字节内码，此方案是将GB2312－80规定的国标码的每字节中最高位置“1”作为汉字机器内码。

CC－DOS所提供的键盘管理模块、显示管理模块、字库管理模块和打印管理模块均是以文件的形式存放在磁盘上。在CC－DOS运行时，它与原来的BIOS一起处理汉字信息。其中键盘管理模块的功能主要是从键盘上输入汉字外码转换成机器的内码。其工作流程可以用下框表描述：



从框表可以看出，汉字输入过程可分为代码识别和代码转换两部分。因此代码识别是前提，而代码转换是必不可少的一步。下面的几个BASIC语言程序就是直接把汉字代码转换成机器代码或者是把ASCⅡ码转换成汉字代码的例子。也就是说，以下的几个程序实际上是把键盘模块的部分功能用BASIC语言程序实现。故此程序的设计思想及方法不再赘述，只是将关键性的语句作一些说明。

［例1］把汉字转换为机器内码。前提是，CC－DOS汉字系统已装入内存。当运行此程序时，每输入一汉字就求出它的内码。程序清单如下：

```
100 INPUT "请输入汉字";A$
120 FOR I=1 TO 2
130 A(I)=ASC((MID$(A$,I,1)))
140 NEXT
150 PRINT "第一内码=";HEX$(A(1))
160 PRINT "第二内码=";HEX$(A(2))
170 PRINT:PRINT
180 GOTO 100
```

RUN
请输入汉字？大［回车］
第一内码＝B4
第二内码＝F3
请输入汉字？

程序中的循环体语句130分别把汉字左右部分转换成ASCⅡ；语句150和语句160分别把相应的ASCⅡ码转换为十六进制的汉字内码。

［例2］已知汉字内码求对应的汉字。程序清单及结果如下：

```
100 CLS
120 DIM A$(20)
130 D$="0123456789ABCDEF"
140 FOR I=0 TO 15 :A$(I)=MID$(D$,I+1,1):NEXT I
150 INPUT "请输入第一内码";A$
160 INPUT "请输入第二内码";B$
170 AA$=LEFT$(A$,1):BB$=RIGHT$(A$,1)
180 CC$=LEFT$(B$,1):DD$=RIGHT$(B$,1)
```

```
190 FOR J=0 TO 15 :IF AA$=A$(J) THEN A=J
200 IF BB$=A$(J) THEN B=J
210 IF CC$=A$(J) THEN C=J
220 IF DD$=A$(J) THEN D=J
230 NEXT J
240 AA=A*16+B:BB=C*16+D
250 PRINT
260 PRINT CHR$(AA);CHR$(BB)
270 PRINT:PRINT
280 GOTO 150
```

RUN [回车]

请输入第一内码? B4 [回车]

请输入第二内码? F3 [回车]

大

语句260把对应汉字内码的汉字输出。

[例3] 此程序可把《GB2312－80》区位编码书打印出来。程序清单如下：

```
100 CLS
110 WIDTH "LPT1:",255
130 FOR I=161 TO 247
140 IF I>167 AND I<176 THEN 300
150 LPRINT " ":LPRINT" "
160 N=I-160:N$=STR$(N)
170 A$=LEFT$(N$,2):B$=RIGHT$(N$,1)
180 IF N<10 THEN A$="0"
190 A=VAL(A$):B=VAL(B$)
200 LPRINT CHR$(163);CHR$(176++A);CHR$(163);CHR$(176+B);"
区 00 01 02 03 04 05 06 07 08 09 10 12 13 14 15 16 17 18 19"
210 FOR J=161 TO 254
220 IF J=161 THEN LPRINT " 0-19 ";
230 LPRINT CHR$(I);CHR$(J);" ";
240 A=J-160
250 IF A=19 THEN LPRINT " ":LPRINT" 20-39 ";
260 IF A=39 THEN LPRINT " ":LPRINT" 40-59 ";
270 IF A=59 THEN LPRINT " ":LPRINT" 60-79 ";
280 IF A=79 THEN LPRINT " ":LPRINT" 80-94 ";
290 NEXT J
300 NEXT I
```

RUN [回车]

·

·

·

·

·

```
16区 00 01 02 03 04 05 06 07 08 09 10 12 13 14 15 16 17 18 19
0-19  啊阿埃挨哎唉哀皑癌蔼矮艾碍爱隘鞍氨安俺
20-39 按暗岸胺案肮昂盎凹敖熬翱袄傲奥懊澳芭捌扒
40-59 叭吧笆八疤巴拔跋靶把耙坝霸罢爸白柏百摆佰
60-79 败拜稗斑班搬扳般颁板版扮拌伴瓣半办绊邦帮
80-94 梆榜膀绑棒磅蚌镑傍谤苞胞包褒剥

17区 00 01 02 03 04 05 06 07 08 09 10 12 13 14 15 16 17 18 19
0-19  薄雹保堡饱宝抱报暴豹鲍爆杯碑悲卑北辈背
20-39 贝钡倍狈备惫焙被奔苯本笨崩绷甭泵蹦迸逼鼻
40-59 比鄙笔彼碧蓖蔽毕毙毖币庇痹闭敝弊必辟壁臂
60-79 避陛鞭边编贬扁便变卞辨辩辫遍标彪膘表鳖憋
80-94 别瘪彬斌濒滨宾摈兵冰柄丙秉饼炳
```

·

·

·

·

该程序的核心思想与例2一致，只要细心观察本例230语句与上例260语句，并注意到该程序的230语句的变量“I”及“J”的作用就能得出结论。

例4 此程序可把国标编码本打印出来。清单及结果如下：

```
1000 CLS
1100 WIDTH "LPT1:",255
1200 LPRINT CHR$(27);CHR$(73);CHR$(73)
1300 Q=0:M=0:N=0
```

```
1400 FOR I=161 TO 247
1500 IF I>167 AND I<167 THEN 210
1600 FOR J=161 TO 254
1700 Q=Q+1
1800 LPRINT CHR$(I);CHR$(J);" ";
1900 A$=HEX$(I-128):B$=HEX$(J-128)
2000 A$="00"+A$:B$="00"+B$
2010 A$=RIGHT$(A$,2):B$=RIGHT$(B$,2)
2020 A1$=LEFT$(A$,1):A2$=RIGHT$(A$,1)
2030 B1$=LEFT$(B$,1):B2$=GIGHT$(B$,1)
2040 LPRINT A1$;" ";A2$;" ";B2$;
2050 IF Q=6 THEN Q=0:N=N+1:LPRINT:GOTO 1700
2060 LPRINT "  ";
2070 IF N=27 AND N/5=INT(N/5) THEN C=2 ELSE C=0
2080 IF N=27 THEN N=0:N=N+1:LPRINT TAB(35)"---";N;
     "---":FOR Z=1 TO 21+C:LPRINT:NEXT
2090 NEXT J
2100 NEXT I
```

RUN

```
J
| .
  .
  .
| 2 1 1  、2 1 2  。2 1 3  · 2 1 4  ˉ 2 1 5  ˇ 2 1 6
ˇ 2 1 6  ¨ 2 1 7  〃2 1 8  々2 1 9  —2 1 A  ～2 1 B
～2 1 B  ‖ 2 1 C  …2 1 D  ' 2 1 E  ' 2 1 F  " 2 1 0
" 2 1 0  " 2 1 1  〔2 1 2  〕2 1 3  〈2 1 4  〉2 1 5
〉2 1 5  《2 1 6  》2 1 7  「2 1 8  」2 1 9  『2 1 A
『2 1 A  』2 1 B  〖2 1 C  〗2 1 D  【2 1 E  】2 1 F
】2 1 F  ±2 1 0  ×2 1 1  ÷2 1 2  ∶2 1 3  ∧2 1 4
∧2 1 4  ∨2 1 5  ∑2 1 6  ∏2 1 7  ∪2 1 8  ∩2 1 9
∩2 1 9  ∈2 1 A  ∷2 1 B  √2 1 C  ⊥2 1 D  ∥2 1 E
∥2 1 E  ∠2 1 F  ⌒2 1 0  ⊙2 1 1  ∫2 1 2  ∮2 1 3
∮2 1 3  ≡2 1 4  ≌2 1 5  ≈2 1 6  ∽2 1 7  ∝2 1 8
∝2 1 8  ≠2 1 9  ≮2 1 A  ≯2 1 B  ≤2 1 C  ≥2 1 D
≥2 1 D  ∞2 1 E  ∵2 1 F  ∴2 1 0  ♂2 1 1  ♀2 1 2
♀2 1 2  °2 1 3  ′2 1 4  ″2 1 5  ℃2 1 6  $2 1 7
$2 1 7  ¤2 1 8  ¢2 1 9  £2 1 A  ‰2 1 B  §2 1 C
§2 1 C  №2 1 D  ☆2 1 E  ★2 1 F  ○2 1 0  ●2 1 1
●2 1 1  ◎2 1 2  ◇2 1 3  ◆2 1 4  □2 1 5  ■2 1 6
■2 1 6  △2 1 7  ▲2 1 8  ※2 1 9  →2 1 A  ←2 1 B
←2 1 B  ↑2 1 C  ↓2 1 D  〓2 1 E    2 2 1    2 2 2
   2 2 2     2 2 3     2 2 4     2 2 5     2 2 6     2 2 7
   2 2 7     2 2 8     2 2 9     2 2 A     2 2 B     2 2 C
   2 2 C     2 2 D     2 2 E     2 2 F     2 2 0  ⒈ 2 2 1
⒈ 2 2 1  ⒉ 2 2 2  ⒊ 2 2 3  ⒋ 2 2 4  ⒌ 2 2 5  ⒍ 2 2 6
⒍ 2 2 6  ⒎ 2 2 7  ⒏ 2 2 8  ⒐ 2 2 9  ⒑ 2 2 A  ⒒ 2 2 B
⒒ 2 2 B  ⒓ 2 2 C  ⒔ 2 2 D  ⒕ 2 2 E  ⒖ 2 2 F  ⒗ 2 2 0
⒗ 2 2 0  ⒘ 2 2 1  ⒙ 2 2 2  ⒚ 2 2 3  ⒛ 2 2 4  ⑴ 2 2 5
⑴ 2 2 5  ⑵ 2 2 6  ⑶ 2 2 7  ⑷ 2 2 8  ⑸ 2 2 9  ⑹ 2 2 A
⑹ 2 2 A                  --- 1 ---
 .
 .
 .
```

缩微检索投影仪及其联机系统研制成功

南京军区科技办公室于4月10日至11日在扬州市主持召开了《缩微检索投影仪及其联机系统》技术鉴定会。

经专家鉴定认为，由南京军区指挥自动化工作站设计并与扬州邗江无线电仪器厂联合研制的缩微检索投影仪及其联机系统，采用光点检索，具有检索快速准确、连续工作时间长、投影幅面大、光照度强、画面清晰、功耗低等特点，适合配于IBM－PC系列及北极星等微型计算机，这是在国内首次实现微机控制的缩微胶卷自动检索系统，居国内领先地位和先进水平，为我国缩微胶卷自动检索技术填补了空白。与会专家一致通过该项目的技术鉴定。（宋川）

用APPLE－Ⅱ微机检测噪声中微弱信号

湘潭大学无线电教研室　秦卓仁　崔如春

数字积累平均法对于检测被噪声淹没的重复性信号是一种极为有效的技术。本文介绍如何用普通APPLE－Ⅱ微机实现此种检测的一种方案，阐述了系统的硬件结构及软件编程等有关问题，并报告它的初步应用

一.引　言

众所周知，常规的信号检测方法是以系统的内部噪声强度为其检测极限。即不能检测出比噪声还弱的信号。当实际的有用信号很弱时，就会出现有用信号被噪声淹没的情况。对于这种淹没在噪声中的信号的检测，称为微弱信号检测。

从60年代起，微弱信号检测技术发展很快，并在各方面获得推广应用。目前，比较有效的是相关检测法和同步取样积累平均法。前者是使用锁定放大器，检测信号的相关系数，测得的一般是个直流量，即只能获知信号的幅度和幅度的变化，不能直接测得信号波形。后一种同步取样积累平均法，可直接得到信号波形，因此要比相关检测法所得到的直观。

同步取样积累平均法的基础是巧妙地利用了有用信号的重复性而噪声则为随机性这一根本差别。设混有噪声的信号为：

$$f_i(t)=s_i(t)+n_i(t) \quad ---- (1)$$

式中$s_i(t)$为有用信号，限定为周期性信号或可重复性信号；$n_i(t)$为噪声，其有效值为$n_i(t)=\delta$。若对信号进行同步取样，经m次取样叠加，对有用信号是按振幅相加，其累积结果为

$$S_0(t_i)=\sum_{i=1}^{m}S_i(t_i)=ms_i(t_i)$$

而噪声的各个相应取样值为一随机序列，m次取样叠加的方差为：

$$D\left[\sum_{i=1}^{m}X_i\right]=m\sum_{i=1}^{m}D\left[X_i\right]=m\delta^2$$

再开方得m次取样叠加后的噪声有效值为：

$$\overline{n_0(t)}=m\delta=\sqrt{m}\,\overline{n_i(t)}$$

故经m次叠加后的电压信噪比为：

$$\frac{S_0(t_i)}{n_0(t)}=\frac{mS_i(t_i)}{\sqrt{m}\;n_i(t)}$$

$$=\sqrt{m}\,\frac{S_i(t)}{n_i(t)}\cdots\cdots(2)$$

可见，经m次同步叠加平均后输出的信噪比改善了$\sqrt{m}$倍。

同步取样积累平均法的关键是必须实现同步取样，即必须满足取样脉冲与待测信号在时间上严格锁定这一关系。另外，取样脉冲的重复频率f_s必须高于待测信号最高频率f_m的一倍以上，即满足取样定理要求的$f_s\geqslant 2f_m$。

对每次的检测样值既可进行模拟处理，也可进行数字处理。随着微型计算机的普及，利用微型机这一强有力的通用数字处理设备，实现多点同步取样，并进行数字累积平均，使微机成为从噪声中检测微弱信号的有力工具的想法和努力，显然是很有意义并值得提倡的。国内已有人在这方面进行了努力，但是都以单板机进行的。作者试用APPLE－Ⅱ机实现此种检测，以使编程和修改较为容易，并便于对结果作进一步的各种数学处理。

二.系统的硬件配置

系统的硬件配置框图如图1。图1中的模数和数模转换采用专配APPLE－Ⅱ机的12位商品模板。极性变换器、前置放大器与低通滤波器则是自

行设计组装的。

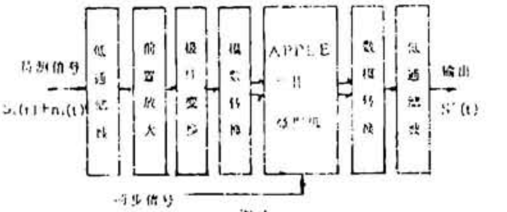


图 1

低通滤波器采用巴特沃斯（Butterworth）特性的低通有源滤波器，如图2。3db截止频率选为系统取样频率的一半，以尽量压缩输入噪声，改善输入信噪比，减少需要的总平均时间。

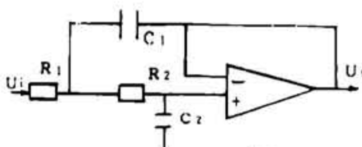


图 2

图3为前置放大器与极性变换电路。当输入信号由−0.1V～+0.1V变化时，极性变换电路的输出则由0～+10V单极性变化，以适应模数转换电路对输入信号的要求。

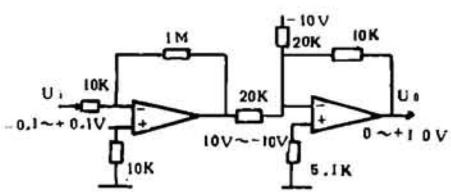


图 3

12位AD−DA转换卡插入APPLE−Ⅱ机I/O设备插座的第7槽。该AD−DA卡的一次AD转换时间约60 μS。实现同步采样所须的外同步脉冲从APPLE−Ⅱ机的录音机输入接口孔加入。当CPU检测有外同步信号时，即用指令启动A/D转换。同步信号一般须先经整形，整形电路可采用通常的施米特电路。

经模数转换后的信号，以数字量的形式存入计算机，经叠加平均后经DA转换和低通滤波器平滑后输出。输出的模拟信号可接至记录仪或示波器。

三，系统的软件编程

程序用6502汇编语言编写，以使运行时间尽量短。程序框图如图4，具体程序见文末附录，这里只作些说明。

1. 程序中设置每次采样点数为80个点，每一次采样过程约13ms左右（程序中加进适当延时），即采样频率在6kHz左右。程序设置的采样累计次数m为一个字节，即2^8=256次，故信噪比能得到$\sqrt{256}=16$倍的改善。

2. 按所用的12位AD转换板说明书，AD转换是通过软开关实现的。即若插于I/O插座第7槽，则可依次用指令访问$C0F4和$C0F5地址各7次，就完成一次AD转换。

3. 程序中用三个存贮单元存放每个采样点的数据。数据的叠加采用变址寻址方式，先存低位后存高位，顺序将各采样点的数值存完一遍后，又从头来，一遍叠于一遍之上。叠加时低位的进位先存于进位位C，再进入高位。

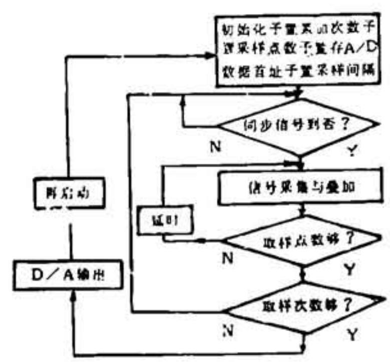


图 4

4. 延时子程序是为控制两个采样点间的时间间隔用的，若要提高或降低采样频率，可去掉或延长这个子程序。

5. 由程序检查特定地址单元$C060（录音输入插口地址）的内容来实现同步。当送入插口的同步信号为正时，（$C060）>128，为负时（$C060）<128。在（$C060）<128时，程序对它进行监视，一旦（$C060）>128，CPU立即发指令启动AD转换。

6. 在启动之后，一次采样累积平均过程完毕，就送出显示，并立即重新开始采样，不断重复

上述过程。但如启动后，有键盘命令发出（压下任何一键）则停止重新采样，改为不断重复送出已采样好了的信号，以供仔细研究（实际上相当于作存贮示波器使用）。如你再按下键盘发出命令，则又回到不断重新采样、显示循环。

四、初步应用及几个有关问题

将上述装置应用于一些实验测量，结果是相当令人满意的，现摘要报告于后：

1. 用于微波吸收的测量

框图如图5。将反射速调管的反射极调制电压作为APPLE—Ⅱ累积平均器的取样同步信号。在吸收很深时，微波检波器的输出接至示波器看到的只是一片噪声，如图6（a）；经数字累积平均后，则如图6（b）。可见，效果十分明显，确实从被噪声淹没无法辨认的信号中清晰地再恢复出来方波信号。

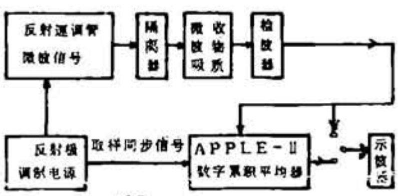


图 5

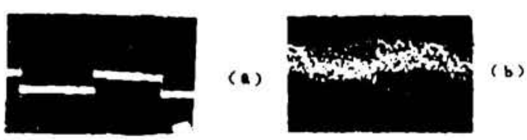


图 6

2. 用于光声信号检测

用于光声信号检测的框图如图7，而图8（a）和（b）则是同步取样累积平均处理前后的波形。

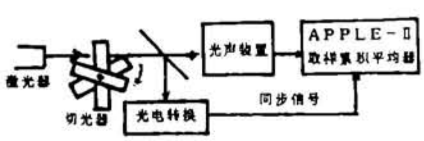


图 7

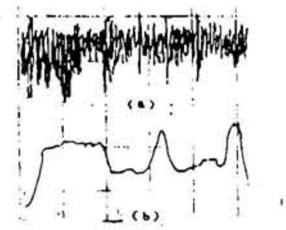


图 8

3. 几个有关问题：

本程序AD数据存贮单元采用三个字节，故对12位AD转换板，不溢出的最大可能叠加次数为 $2^{24}/2^{12}=4096$ 次。但本程序叠加次数只为256次，故只须适当修改程序，就可使信噪比的改善进一步提高。

本程序使用的是线性叠加平均，如须改为指数平均或归一化平均，可参考有关资料进行编程。

模数转换前的总增益设置，必须使噪声峰值不超过满刻度，即不能使之过载，否则将使输出信号产生抖动。

同步信号与待处理信号，在时间上必须严格锁定，否则将使叠加平均后的波形模糊不清。

待处理信号在一遍采样循环过程中，相对颤动大（重复性差），也将使叠加平均结果恶化。

应充分利用前置输入滤波器抑制掉信号的带外噪声，这可改善输入信噪比，并降低总的叠加平均时间。如果经前置滤波器能使输入噪声降低10dB，则达到给定信噪比的平均时间减少为只是原来的十分之一。

五、结　论

实践证明，利用APPLE－Ⅱ微机以数字叠加平均法，从噪声中提取微弱信号是可行的，而且比用Z80等单板机来实现此种工作更为灵活方便。

程序清单：
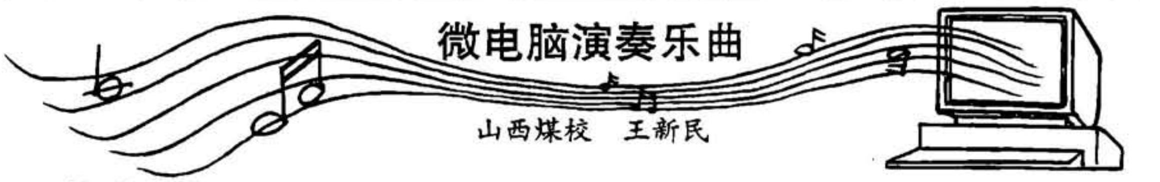
```
FILE NAME IS SAMPLE.OBJ0
          ORG   $4400
FIRST:    LDA   #$FD
          STA   $304
          LDA   $C010
          LDA   #0
          TAX
          LDA   #240
          STA   $302
NOUGHT:   LDA   #0
          STA   $5000,X
```

```
                INX
                DEC     $302
                BNE     NOUGHT
                LDA     #$FF
                STA     $300
        BEGIN:  LDA     #$50
                STA     $301
                LDA     #0
                TAX
        SIGNAL  LDA     #$1
                STA     $C0F0
                LDA     $C0F3
                LDA     $C0F4
                NOP
                LDA     $C0F4
                LDA     $C0F4
                LDA     $C0F4
                LDA     $C0F4
                LDA     $C0F4
                LDA     $C0F4
                LDA     $C0F5
                LDA     $C0F5
                LDA     $C0F5
                LDA     $C0F5
                LDA     $C0F5
                LDA     $C0F5
                LDA     $C0F5
                LDA     $C0F1
                CLC
                ADC     $5000,X
                STA     $5000,X
                INX
                LDA     $C0F2
                AND     #$0F
                ADC     $5000,X
                STA     $5000,X
                INX
                LDA     $5000,X
                ADC     #$0
                STA     $5000,X
                INX
                DEC     $301
                BNE     DELAY
                DEC     $300
                BEQ     AVERAG
                JMP     SYNC
        DELAY:  LDA     #1
                JSR     $FCA8
                JMP     SIGNAL
        SYNC:   LDA     $C060
                CMP     #128
                BPL     SYNC
        OUT1:   LDA     $C060
                CMP     #128
                BPL     BEGIN
                BMI     OUT1
        AVERAG: LDA     #0
                TAX
                LDA     #80
                STA     $303
        OUT2:   INX
                INX
                LDA     $5000,X
                STA     $C0F7
                DEX
                LDA     $5000,X
                STA     $C0F6
                INX
                INX
                DEC     $303
                BNE     OUT2
                DEC     $304
                BEQ     DELAY2
                LDA     #$20
                JSR     $FCA8
                JMP     AVERAG
        DELAY2: LDA     #$1A
                STA     $305
        OUT3:   JSR     $FCA8
                INC     $4E
                BNE     KEY
                INC     $4F
        KEY:    BIT     $C000
                BMI     CONTU
                DEC     $305
                BNE     OUT3
                JMP     FIRST
        CONTU:  LDA     #10
                STA     $308
        OUT6:   JSR     $FCA8
                DEC     $308
                BNE     OUT6
        DISPLAY LDA     #$FF
                STA     $306
                LDA     $C010
        AVERGE2: LDA    #0
                TAX
                LDA     #80
                STA     $303
        OUT4:   INX
                INX
                LDA     $5000,X
                STA     $C0F7
                DEX
                LDA     $5000,X
                STA     $C0F6
                INX
                INX
                DEC     $303
                BNE     OUT4
                LDA     #0
                LDA     #30
                JSR     $FCA8
                DEC     $306
                BNE     AVERGE2
                INC     $4E
                BNE     KEY2
                INC     $4F
        KEY2:   BIT     $C000
                BPL     DISPLAY
                JMP     FIRST

ASSEMBLY: NO ERRORS
```

(上接第35页)

```
": PRINT "      1 2 3 4 5 6 7 8 9 *
" -- - ↑": END
4  FOR J = I TO I + I1: READ A: POKE J,A: NEXT :I = J: FOR J = I TO I + I2: POKE J,0: NEXT :I = J: RETURN
5  FOR J = I TO I + I2: POKE J,0: NEXT :I = J: FOR J = I TO I + I1: READ A: POKE J,A: NEXT :I = J: RETURN
6  DATA 4,14,4,0,14,0,8,12,8,8,28,12,18,8,4,30,30,16,8,18,12,16,24,20,62,16,30,2,14,16,14,12,2,14,18,12,30,16,8,4,4,12,18,12,18,12,12,18,28,16,12,68,238,68,0,238,0
7 D$ = CHR$ (4): FOR I = 768 TO 809: READ C: POKE I,C: NEXT : CALL 768: PRINT D$;"BSAVESETG,A$8D00,L$300": CALL 797: PRINT D$;"BSAVESETF,A$8D00,L$300": PRINT D$;"BLOADSETA"
8  DATA 169,0,168,170,185,0,135,153,0,141,200,208,247,232,138,201,3,240,9,238,6,3,238,9,3,76,4,3,96,169,138,141,6,3,169,141,141,9,3,76,0,3
```



微电脑演奏乐曲

山西煤校　王新民

利用苹果机主机箱内的喇叭，让微电脑唱歌，是很容易办到的。这儿所说的唱歌，只是奏出乐曲，并唱不出歌词。即使是演奏乐曲，对青少年学生来说也是有着很强的吸引力的。青少年学生的求知欲很强，在辅导他们上机时，给他们讲解一些知识性、趣味性融为一体的内容，他们是很乐意接受的。

我曾在辅导学生上机时给他们示范过演奏一些乐曲的程序。听听微电脑演奏乐曲，是很有意思的。下面将电影插曲《月饼歌》的程序清单列出来，与读者共欣赏。

程序清单如下：

```
]LIST

5  REM  Yue bing ge
10  HOME : FOR I = 1 TO 5: PRINT : NEXT
    I
15  PRINT "                    Yue bing ge":
     PRINT
20  PRINT "                  SHAN XI MEI XIA
    O": PRINT
25  PRINT "                  Wang    1986.5.1
    2"
30  FOR I = 0 TO 20: READ N: POKE I + 7
    70,N: NEXT
35  DATA 173,48,192,136,208,5,206,1,3,2
    40,9,202,208,245,174,0,3,76,2,3,96
40  READ YG,YC
45  IF YC = 0 THEN  END
50  POKE 768,YG: POKE 769,YC: CALL 770
55  GOTO 40
60  DATA  128,70,128,30,128,30,128,70,1
    28,70,144,160,152,70,172,70,128,70,
    128,30,128,30,128,70,129,70,144,160
    .152,70,172,70,228,70,192,70,228,70
    ,172,70,172,70,192,70,228,160,255,2
    55,228,70,255,160,1,160
65  DATA 172,70,172,70,172,70,172,70,15
    2,70,128,70,192,70,228,70,172,228,1
    52,70,172,255,1,70,228,70,172,70,17
    2,70,192,70,228,160,192,255,172,70
70  DATA 192,255,1,70,228,70,172,160,19
    2,70,172,70,152,70,172,70,192,70,25
    5,70,228,70,172,70,192,70,228,160,2
    28,70,1,70,228,70,192,70,228,70
75  DATA 172,70,172,70,192,70,228,160,2
    55,255,228,70,255,255,1,70,0,0
```

程序说明

1. 5～15语句是注释、清屏、在显示屏的中上位置显示出歌曲名。

2. 30、35语句把机器语言子程序储入内存，为安全起见，通常是存放在770号开始的单元中。机器语言中不能使用变量名，变量的值必须经由POKE语句送入。这个子程序一般是演奏活泼欢快、节奏感强的乐曲。

3. 40语句把数据区存放的音高与音长读入变量YG、YC，50语句再把它们POKE到记忆体768、769两个位置，最后利用770的机器发音子程序演奏出来。

4. 音高和音长的值分别按表1和表2选取。

表1

音阶	0	$\underset{\cdot}{5}$	$\underset{\cdot}{6}$	$\underset{\cdot}{7}$	1	2	3	4	5	6	7	$\dot{1}$	$\dot{2}$	$\dot{3}$
数值	1	255	228	204	192	172	152	144	128	114	102	96	88	78

表2

节拍	$\frac{1}{4}$拍	$\frac{1}{2}$拍	$\frac{3}{4}$拍	1拍	$1\frac{1}{2}$拍
数值	30	70	110	160	255

中音1（do）取192，高音$\dot{1}$（do）取96（=192／2），音阶每提高八度，数值减小一半，音阶每降低八度，数值增加一倍。音高和音长的控制数值只能在0到255的范围内。因此低于$\underset{\cdot}{5}$和长于一拍半的声音就不能实现，只好把低音提高八度、把两拍改为两个一拍。

5. 要想让微电脑演奏别的乐曲，只需将乐曲的音阶和音高相应的数据存放在数据区60语句之后的DATA语句中。